



(19)

Europäisches Patentamt
European Patent Office
Office européen des brevets



(11)

EP 1 091 303 A3

(12)

EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
05.02.2003 Bulletin 2003/06

(51) Int Cl.7: G06F 17/27

(43) Date of publication A2:
11.04.2001 Bulletin 2001/15

(21) Application number: 00308681.6

(22) Date of filing: 03.10.2000

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE

Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 05.10.1999 US 412822

(71) Applicant: MICROSOFT CORPORATION
Redmond, WA 98052 (US)

(72) Inventors:

- Reynar, Jeffrey C.
Woodinville, Washington 98072 (US)

- Rucker, Erik
Seattle, Washington 98105 (US)
- Kim, Paul Kyong Hwan
Seattle, Washington 98107 (US)

(74) Representative:
Wright, Howard Hugh Burnby et al
Withers & Rogers,
Goldings House,
2 Hays Lane
London SE1 2HW (GB)

(54) Method and system for providing alternatives for text derived from stochastic input sources

(57) A computer-implemented method for providing a candidate list of alternatives for a text selection containing text from multiple input sources, each of which can be stochastic (such as a speech recognition unit, handwriting recognition unit, or input method editor) or non-stochastic (such as a keyboard and mouse). A text component of the text selection may be the result of data processed through a series of stochastic input sources, such as speech input that is converted to text by a speech recognition unit before being used as input into an input method editor. To determine alternatives for the text selection, a stochastic input combiner parses the text selection into text components from different input

sources. For each stochastic text component, the combiner retrieves a stochastic model containing alternatives for the text component. If the stochastic text component is the result of a series of stochastic input sources, the combiner derives a stochastic model that accurately reflects the probabilities of the results of the entire series. The combiner creates a list of alternatives for the text selection by combining the stochastic models retrieved. The combiner may revise the list of alternatives by applying natural language principles to the text selection as a whole. The list of alternatives for the text selection is then presented to the user. If the user chooses one of the alternatives, then the word processor replaces the text selection with the chosen candidate.

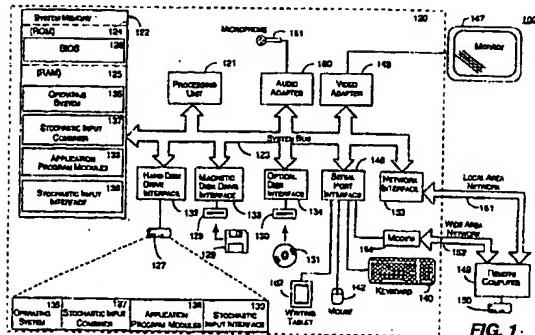


FIG. 1.



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 00 30 8681

DOCUMENTS CONSIDERED TO BE RELEVANT			CLASSIFICATION OF THE APPLICATION (Int.Cl.7)	
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	G06F17/27	
X	US 5 502 774 A (BELLEGARDA JEROME R ET AL) 26 March 1996 (1996-03-26)	1,3-5, 12,13, 17,18,28	G06F17/27	
Y	* abstract * * column 3, line 21 - line 31; figure 1 * * column 3, line 51 - line 67 * * column 5, paragraph 3 * * column 9, paragraph 3; figure 2 * ---	2,6-8, 10,11, 14-16, 19,20, 25,27		
X	US 5 864 805 A (OLSEN PEDER ANDREAS ET AL) 26 January 1999 (1999-01-26)	21-24,26		
Y	* abstract * * column 3, line 11 - line 20 * * column 3, line 65 - line 66; figure 2 * * column 4, line 8 - line 14 * * column 4, line 30 - line 35; figure 3 * * column 4, line 51 - line 54 * * column 5, line 29 - line 56; figures 4,5 *	25,27	TECHNICAL FIELDS SEARCHED (Int.Cl.7)	
Y	US 5 680 511 A (YAMRON JONATHAN P ET AL) 21 October 1997 (1997-10-21) * abstract * * column 8, line 65 - column 9, line 3; figure 1 * * column 10, line 27 - line 30 * * column 12, line 49 - column 13, line 4 * * column 13, line 55 - column 14, line 2 * * column 17, line 17 - line 27 * ---	2,6,7	G06F	
		-/-		
The present search report has been drawn up for all claims				
Place of search	Date of completion of the search	Examiner		
BERLIN	6 December 2002	Barieux, M		
CATEGORY OF CITED DOCUMENTS		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons A : technological background O : non-written disclosure P : intermediate document B : member of the same patent family, corresponding document		
EPO FORM 1503.05.92 (PA/CON)				



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 00 30 8681

DOCUMENTS CONSIDERED TO BE RELEVANT			CLASSIFICATION OF THE APPLICATION (Int.Cl.)
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.)
Y	KONNO A ET AL: "Postprocessing algorithm based on the probabilistic and semantic method for Japanese OCR" DOCUMENT ANALYSIS AND RECOGNITION, PROCEEDINGS OF THE SECOND INTERNATIONAL CONFERENCE ON TSUKUBA SCIENCE CITY, JAPAN, IEEE COMPUT. SOC, 20 - 22 October 1993, pages 646-649, XP010135698 LOS ALAMITOS, CA, USA ISBN: 0-8186-4960-7 * abstract * * page 647, right-hand column, line 7 - line 22; figure 2 * * page 648, left-hand column, line 4 - line 22; figure 3 * * page 648, right-hand column, line 5 - line 10 * ---	8,10,11	
Y	EP 0 570 660 A (IBM) 24 November 1993 (1993-11-24) * abstract * * page 3, line 43 - line 51 * * page 5, line 47 - line 50; figure 1 * * page 6, line 1 - line 7 * ---	14-16, 19,20	TECHNICAL FIELDS SEARCHED (Int.Cl.)
A	US 5 852 801 A (CHOW YEN-LU ET AL) 22 December 1998 (1998-12-22) * abstract * * column 7, line 59 - line 63; figure 2 * * column 8, paragraph 3 - column 9, paragraph 1; figures 3,4 * -----	1,3,13, 17-21, 26-28	
The present search report has been drawn up for all claims			
Place of search		Date of completion of the search	Examiner
BERLIN		6 December 2002	Barieux, M
CATEGORY OF CITED DOCUMENTS		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document	
X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background D : non-written disclosure P : intermediate document			

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 00 30 8681

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

06-12-2002

Patent document cited in search report		Publication date		Patent family member(s)	Publication date
US 5502774	A	26-03-1996	US	5621809 A	15-04-1997
US 5864805	A	26-01-1999	NONE		
US 5680511	A	21-10-1997	DE	69616544 D1	06-12-2001
			EP	0830668 A1	25-03-1998
			WO	9641333 A1	19-12-1996
EP 0570660	A	24-11-1993	US	5293584 A	08-03-1994
			CA	2091912 A1	22-11-1993
			DE	69315374 D1	08-01-1998
			DE	69315374 T2	28-05-1998
			EP	0570660 A1	24-11-1993
US 5852801	A	22-12-1998	NONE		

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-249920

(P2001-249920A)

(43)公開日 平成13年9月14日 (2001.9.14)

(51) Int.Cl.⁷
 G 06 F 17/24
 3/16
 17/22
 G 10 L 15/18

識別記号
 5 5 4
 3 4 0
 5 0 3
 5 0 6

F I
 C 06 F 17/24
 3/16
 17/22
 C 10 L 3/00
 5 3 7 Z

マーク*(参考)

審査請求 未請求 請求項の数27 OL 外国語出願 (全 82 頁) 最終頁に続く

(21)出願番号

特願2000-305651(P2000-305651)

(22)出願日

平成12年10月5日 (2000.10.5)

(31)優先権主張番号 09/412822

(32)優先日 平成11年10月5日 (1999.10.5)

(33)優先権主張国 米国(US)

(71)出願人 595124929

マイクロソフト コーポレイション
MICROSOFT CORPORATION
アメリカ合衆国 98052-6399 ワシントン州
レドマンド ワン マイクロソフト ウェイ (番地なし)(72)発明者 ジェフリー スイー、レイナー
アメリカ合衆国 98072 ワシントン州
ウッディンヴィル エヌイー ワンハンド
レッドサーティエイス ウェイ 23322

(74)代理人 100083806

弁理士 三好 秀和 (外1名)

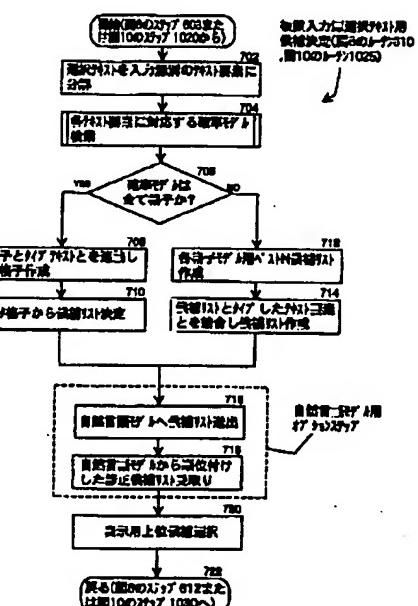
最終頁に続く

(54)【発明の名称】 推測入力源からのテキストに対して候補を提供する方法およびシステム

(57)【要約】

【課題】 選択テキストに対して候補リストを提供する方法を提供する。

【解決手段】 選択テキスト用の候補を決定するに当たり、推測入力結合器は、選択テキストを、入力源ごとのテキスト要素に分解する。推測入力結合器は、各テキスト要素の候補を含んだ確率モデルを検索する。テキスト要素が連続した推測入力源の処理結果であれば、推測入力結合器は、全入力源の結果の確率を正確に反映した確率モデルを取得する。推測入力結合器は、検索した確率モデルを結合することにより、選択テキスト用の候補リストを作成する。推測入力結合器は、選択テキスト全体に自然言語原理を適用することにより、候補リストを修正することができる。選択テキスト用候補リストは、ユーザーに提示する。ユーザーが候補の1つを選ぶと、ワードプロセッサはその選んだ候補で選択テキストを置換する。



【特許請求の範囲】

【請求項1】 入力源を異にする複数のテキスト要素で構成した選択テキストを受け取るステップと、前記テキスト要素の少なくとも1つが、推測入力源または連続した推測入力源からの推測テキスト要素であり、前記選択テキスト用の候補を表示するコマンドを受け取るステップと、前記選択テキストを前記テキスト要素に分解するステップと、前記推測テキスト要素に対応する推測入力源または連続した推測入力源から前記推測テキスト要素に対応する確率モデルを取り出すステップと、前記確率モデルと他のテキスト要素とを結合して前記選択テキスト用候補リストを作成するステップと、前記選択テキスト用候補リストを表示装置に表示するステップとを有することを特徴とするコンピュータで実施するテキスト修正方法。

【請求項2】 前記表示した候補の1つを選択するユーザコマンドを受け取るステップと、前記選択テキストを前記選択した候補で置き換えるステップとをさらに有することを特徴とする請求項1に記載の方法。

【請求項3】 前記選択テキストに対する編集を受け取るステップと、編集した前記選択テキスト用の修正候補リストを作成するステップと、前記編集した選択テキスト用修正候補リストを表示するステップとをさらに有することを特徴とする請求項1に記載の方法。

【請求項4】 前記推測テキスト要素の1つに対する編集を受け取るステップと、編集した前記推測テキスト要素に対応する推測入力源または連続した推測入力源から当該編集した推測テキスト要素用の改訂確率モデルを取り出すステップと、前記改訂確率モデルと前記選択テキストに対応する他の確率モデルとを結合して、編集した前記選択テキスト用の修正候補リストを作成するステップと、前記編集した選択テキスト用修正候補リストを表示するステップとをさらに有することを特徴とする請求項1に記載の方法。

【請求項5】 前記選択テキストが、ワードプロセッサとスプレッドシートとブラウザと電子メールプログラムと音楽編曲プログラムとCADプログラムとプレゼンテーションプログラムとオペレーティングシステムとからなるグループから選択した1アプリケーションのファイル内におけるテキストの一部を有することを特徴とする請求項1に記載の方法。

【請求項6】 前記選択テキスト用候補を表示するステップが、前記選択テキスト用候補を確率順に並べるステップと、

前記候補を前記確率順で表示装置に表示するステップとをさらに有することを特徴とする請求項1に記載の方法。

【請求項7】 前記候補を前記確率順で表示するステップが、所定数の上位候補を選択するステップと、前記選択した上位候補を表示装置に表示するステップとをさらに有することを特徴とする請求項6に記載の方法。

【請求項8】 前記選択テキストが複数の推測テキスト要素からなり、前記確率モデルを結合するステップが、各推測テキスト要素用の確率モデルを結合して、前記選択テキスト用の中間候補リストを作成するステップと、前記中間候補リストを自然言語モデルへ送るステップと、

前記自然言語モデルは、自然言語原理を前記選択テキスト全体に適用することによって前記中間候補リストを再評価し、前記選択テキスト用の修正候補リストを作成し、該修正候補リストを提供するステップと、前記修正候補リストを前記自然言語モデルから受け取り、それを前記選択テキスト用の候補リストとして表示するステップとをさらに有することを特徴とする請求項1に記載の方法。

【請求項9】 前記修正候補リストが、前記自然言語モデルが作成した追加候補であって前記自然言語モデルへ送った中間候補リストには無い追加候補を含むことを特徴とする請求項8に記載の方法。

【請求項10】 1つ以上の推測テキスト要素用に検索した確率モデルを自然言語モデルへ送り、該確率モデルを自然言語原理に基づく再評価に利用するステップをさらに有することを特徴とする請求項8に記載の方法。

【請求項11】 各推測テキスト要素用の確率モデルを自然言語モデルへ送り、該確率モデルを修正候補リスト作成に利用するステップをさらに有することを特徴とする請求項8に記載の方法。

【請求項12】 前記選択テキストが複数の推測テキスト要素からなり、前記テキスト要素用の確率モデルが格子からなり、

前記確率モデルを結合して前記選択テキスト用の候補リストを作成するステップが、

前記格子を連結して、非推測入力源からの全テキスト要素に関する情報をも含んだメタ格子を作成するステップと、

前記メタ格子から前記選択テキスト用の候補リストを作成するステップとをさらに有することを特徴とする請求項1に記載の方法。

【請求項13】 前記選択テキストが複数の推測テキスト要素からなり、前記確率モデルの1つがベストn候補からなり、他の確率モデルが格子からなり、前記確率モデルを結合して前記選択テキスト用の候補リ

ストを作成するステップが、
前記格子に対応するベストn候補リストを作成するステップと、
前記ベストn候補リストを結合して前記選択テキスト用候補リストを作成するステップとをさらに有することを特徴とする請求項1に記載の方法。

【請求項14】 テキスト要素に対応する確率モデルを推測入力源から取り出す前記ステップが、
前記テキスト要素は連続した推測入力源からのものか否かを決定するステップと、
前記テキスト要素が連続した推測入力源からのものであれば、各推測入力源からの確率モデルを結合して連続確率モデルを作成するステップと、
前記連続確率モデルを前記テキスト要素用確率モデルとして取り出すステップとをさらに有することを特徴とする請求項1に記載の方法。

【請求項15】 前記推測テキスト要素に対応する推測入力源または連続した推測入力源から、前記推測テキスト要素に対応する確率モデルを取り出すステップが、
連続した推測入力源における第1推測入力へのユーザ入力を受け取るステップと、
前記第1推測入力源が生成する複数の候補からなる推測結果を選択するステップと、
前記第1推測入力源が生成した推測結果の各候補を、前記連続した推測入力源における第2推測入力への入力として用いることにより、前記第2推測入力源の複数の推測結果を生成するステップと、
前記第2推測入力源の推測結果のいずれかがベストn候補リストでなければ、その推測結果をベストn候補リストに変換するステップと、

前記第2推測入力源の推測結果を結合して前記第2推測入力源の総合候補リストを作成するステップとをさらに有することを特徴とする請求項1に記載の方法。

【請求項16】 前記第2推測入力源の推測結果を結合して前記第2推測入力源の総合候補リストを作成するステップが、
前記第2推測入力源の複数推測結果の各一意の候補に対して1つの項目を前記総合候補リスト内に生成するステップと、

前記総合候補リスト内の各候補の確率を、前記第2推測入力源の複数推測結果において当該候補に割り当てた全確率を合計して計算するステップと、
前記計算した確率を対応する候補に割り当てるステップとをさらに有することを特徴とする請求項15に記載の方法。

【請求項17】 請求項1記載の方法を実行するためのコンピュータ実行命令を有するコンピュータ読取可能媒体。

【請求項18】 請求項1記載の方法を実行するためのコンピュータ。

【請求項19】 請求項16記載の方法を実行するためのコンピュータ実行命令を有するコンピュータ読取可能媒体。

【請求項20】 請求項16記載の方法を実行するためのコンピュータ。

【請求項21】 ユーザから選択テキストを受け取るステップと、
前記選択テキスト用の候補を表示するコマンドを受け取るステップと、
前記選択テキストを修正範囲モデルへ送って修正範囲を調整すべきか否かを決定するステップと、
前記選択テキストと少なくとも1つの隣接語とを含む単位テキストを、前記修正範囲モデルから受け取るステップと、
前記単位テキスト用の候補リストを作成するステップと、
前記単位テキスト用候補リストを表示装置に表示するステップとを有することを特徴とするテキスト修正をコンピュータで実施する方法。

【請求項22】 前記表示した候補の1つを選択するユーザコマンドを受け取るステップと、
前記単位テキストを前記選択した候補で置き換えるステップとをさらに有することを特徴とする請求項21に記載の方法。

【請求項23】 前記隣接語が、前記選択テキスト内の1語を不正確にしたエラーに起因して、不正確であることを特徴とする請求項21に記載の方法。

【請求項24】 前記修正範囲モデルが、自然言語モデルと類似エラーモデルと音声モデルと手書きモデルと映像モデルとからなるグループから選択した判断基準を含むことを特徴とする請求項21に記載の方法。

【請求項25】 前記単位テキスト用の候補リストを作成するステップが、
前記単位テキストを、入力源別のテキスト要素に分解するステップと、
前記テキスト要素の1つが推測テキスト要素であるか否かを決定するステップと、
前記テキスト要素の1つが推測テキスト要素であれば、その推測テキスト要素用の確率モデルを検索するステップと、

前記確率モデルと他のテキスト要素とを結合して前記単位テキスト用候補リストを作成するステップとをさらに有することを特徴とする請求項21に記載の方法。

【請求項26】 請求項21記載の方法を実行するためのコンピュータ実行命令を有するコンピュータ読取可能媒体。

【請求項27】 請求項25記載の方法を実行するためのコンピュータ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータにテキストを入力するための方法に関する。さらに詳しくは、複数の推測入力源からの選択テキストに対して、複数の候補を提供することに関する。

【0002】

【従来の技術】コンピュータのユーザは、キーボードとマウスを使ってワードプロセッサにテキストを入力してきた。最近のワードプロセッサは改良が進み、ユーザは音声や手書き等の入力方法を介しても、テキストの入力ができる。コンピュータは、このような入力を必ずしも正確に解釈できないが、その入力に対する候補テキストのリストを作成できる。さらにコンピュータは、ユーザの意図に対する確率を各候補に付加できる。このように確率を付加できる入力を「推測入力」と呼ぶ。一方、タップしたテキストのように正確に決定できる入力は、「非推測入力」と呼ぶ。

【0003】

【発明が解決しようとする課題】一般に、推測入力源からワードプロセッサへ入力したテキストは、ユーザが編集しないと意図したテキストにならない。この編集作業は、ユーザが推測テキスト内でテキストを選択し、その選択テキストに対応する候補を要求することで簡単に行われてきた。コンピュータは、候補の要求を受けると、グラフィカルユーザインタフェースを介して、選択テキストに対する候補をユーザに提供する。ユーザが候補の1つを選ぶと、コンピュータはその選択候補で選択テキストを置き換える。

【0004】例えばユーザが、音声等の推測入力源を用い、ワードプロセッサにテキストを入力し、そのテキストを編集する場合を考える。例えばユーザは、テキスト中の1語を置換するため、キーボードを使って新しい語をタブし、コンピュータに入力する。現在のワードプロセッサは、タブしたテキストを、選択テキスト用候補の中に挿入しない。従って、推測入力源からのテキストを編集した場合に、この編集したテキストにも候補を提供する手段が必要である。

【0005】選択テキストが複数の推測入力源を元とする場合、そのような選択テキストに対してユーザが候補を要求すると問題が起きる。例えば選択テキストが、手書き入力に基づく語と音声入力に基づく語とを含み、この選択テキストに対してユーザが候補を要求するような場合である。現在のワードプロセッサは、このような選択テキストに対して、意味のある候補を提供できない。従って、複数の推測入力源に基づく選択テキストにも候補を提供する手段が必要である。

【0006】入力システム(IME)もまた、推測データを生成し、それをワードプロセッサへの入力とする。一般にIMEは、入力を外国語テキストに変換する。IMEへの入力は例えば、キーボードおよびマウスを介してコンピュータに入力したテキストである。IMEは、

アジア言語等における表意文字を生成するのに特に有用である。このような言語は、キーボードのキーよりはるかに多い表意文字を持つ。特定の表意文字をコンピュータに入力するには、複数のキーを打ち込み、それをIMEが合成文字として解釈する。

【0007】例えばユーザは、希望の中国語文字の音声スペルを英字でタイプする。多くの中国語文字は発音が似ており、タイプした音声スペルは異なる中国語文字に対応することが多い。するとIMEは、タイプした音声スペルに対応する最適候補をユーザに提供し、ユーザはその中から正しい1つを選べる。

【0008】プログラマは、IMEに音声を入力できれば役に立つと考えてきた。まず音声をテキストに変換し、そのテキストをIMEへの入力とする。前述したように、音声の解釈は元来が推測である。このため音声解釈から生成したテキストは、ユーザが意図したテキストでないことがある。不正確なテキストをIMEへ入力すれば、IMEは不正確な結果を生成する。従って音声をIMEへの入力として使う場合、音声データを解釈するプログラムは、音声解釈によって生成したテキストをまずユーザに修正させ、その修正テキストをIMEへ入力する。IMEがそれを外国語に翻訳すると、ユーザは再び候補から選択する。IMEの翻訳結果も推測だからである。このように2つの編集処理をユーザに求めるのは、非能率で不便である。従って、IMEへの音声入力処理を行う改良方法が必要である。

【0009】

【課題を解決するための手段】本発明は、テキスト編集を受け入れる推測入力結合器を提供することにより、前記必要を満たす。本発明は、ユーザが選択したテキストに対して候補を提供する。この選択テキストは、1つ以上の推測入力を含む複数の入力源から生成したものでも構わない。

【0010】本発明の推測入力結合器は、グラフィカルユーザインタフェースを介してユーザに候補リストを提供する。ユーザは、その候補から1つを選択し、その候補で選択テキストを置換する。選択テキストとは、編集する目的でユーザが強調表示したテキストである。この方法は、ユーザが自分で候補を考え、それに基づいてキーボードとマウスによって修正を行うより早い。選択テキストに対する希望の候補が見つからなければ、ユーザはキーボードとマウスを使い、選択テキストを編集できる。すると推測入力結合器は、そのユーザの修正を取り入れた候補リストを提供する。ユーザが選択テキストを部分的に編集するだけで、推測入力結合器は候補を提供するから、編集効率が向上する。

【0011】この推測入力結合器は、自然言語モデルも利用できる。自然言語モデルは、自然言語原理を用いて選択テキストを分析し、より良い候補リストを提供できる。これもまた、ユーザの編集効率を向上する。

【0012】本発明は、広くテキスト修正方法に関する。ユーザはまず、コンピュータにテキストを入力する。この時ユーザは、複数の推測入力源や、キーボードおよびマウスを用いる。

【0013】キーボードおよびマウスで入力したテキストは、非推測入力源の例である。すなわちこのテキストは、正確に決定可能である。一方、推測入力源からの入力は、候補リストに変換する。各候補は、正確さの確率が100%以下である。コンピュータは音声を完璧に解釈することはできないため、音声認識装置は推測入力源の一例である。従って音声入力は、テキスト候補リストに変換する。推測入力源の他の例は、手書き認識装置および入力システム（IME）である。選択テキスト中のテキスト要素の入力源が推測入力源である場合、最適候補をそのテキスト要素の代表として使う。

【0014】ユーザは、コンピュータにテキストを入力し、その入力テキストの一部を選択し、修正作業を開始する。選択テキストは、複数のテキスト要素を含む。各テキスト要素は、選択テキストの一部であり、单一の入力源を介してユーザが入力したものである。選択テキストは、様々な入力源からのテキスト要素を含むこともある。それら入力源は、推測入力源も含み得る。

【0015】テキストを選択した後ユーザは、選択テキスト全体に対する候補を表示するコマンドを入力する。推測入力結合器は、選択テキストを入力源別のテキスト要素に分解し、各推測入力源のテキスト要素用の候補を代表する確率モデルを検索する。この確率モデルは、各候補とその確率を含んだリストを持つ。確率モデルは、格子を含むこともできる。

【0016】推測入力結合器は、前記確率モデルと他のテキスト要素とを結合し、前記選択テキスト用候補リストを作成し、そのリストをモニタ等の表示装置に表示する。ユーザは、表示候補の1つを選択できる。選択した候補は、前記選択テキストに置き換わる。

【0017】推測入力結合器は、自然言語モデルを利用できる。この場合、推測入力結合器は、各テキスト要素の確率モデルを結合して選択テキスト用中間候補リストを作成し、それを自然言語モデルへ送る。自然言語モデルは、選択テキスト全体に自然言語原理を適用し、前記中間候補リストを再評価し、修正候補リストを作成する。自然言語モデルは、前記中間候補リストに無い新しい候補を前記修正候補リストに追加できる。自然言語モデルは、修正候補リストを推測入力結合器へ送る。推測入力結合器は、その修正候補リストを表示する。

【0018】本発明の他の形態において、テキスト要素は、連続した推測入力源から生成した推測テキスト要素である。少なくとも1つの推測入力源が推測結果を生成し、それを次の推測入力源の入力とする。第1推測入力源はユーザ入力を受け取り、それに対する候補を生成する。これら候補は、第2推測入力源の入力となる。例え

ば音声認識装置はテキストを生成し、そのテキストがIMEへの入力となる。連続した推測入力源が1つの推測テキスト要素を生成すると、推測入力結合器は連続確率モデルを生成できる。この連続確率モデルは、推測テキスト要素用候補とこれら候補の正確な確率とを含む。これら確率は、連続した入力源の各入力源への入力の確率である。この形態によれば、第1推測入力源から第2推測入力源へ入力を与える際、ユーザが入力を選択する必要がない。

【0019】推測入力結合器は、連続した推測入力源からのテキスト要素に対して連続確率モデルを生成する際、第1推測入力源にユーザ入力を送る。第1推測入力源は、それを処理し、推測結果を生成する。この推測結果は、複数の候補を有する。推測入力結合器は、その推測結果を第2推測入力源への入力とする。第2推測入力源は、複数の第2推測結果を生成する。各第2推測結果は、複数の候補を有する。第2推測結果のいずれかがベストn候補リストでなければ、推測入力結合器は、それをベストn候補リストに変換する。全推測結果を同一フォーマットに変換すれば、結合処理が簡単だからである。推測入力結合器は、第2推測結果を結合し、総合候補リストを作成する。連続した推測入力源が2個であれば、前記総合候補リストは、前記連続した推測入力源からのテキスト要素用確率モデルとなる。

【0020】推測入力結合器は、選択テキスト中の修正範囲を広げ、より大きな単位テキストにすることもできる。これを行うため推測入力結合器は、選択テキストを修正範囲モデルへ送り、修正範囲の調整が必要か否かを判断する。修正範囲モデルは、推測入力結合器に単位テキストを返す。この単位テキストは、選択テキストと少なくとも1つの隣接語とを含む。推測入力結合器は、その単位テキスト用の候補リストを作成し、それを表示装置に表示する。

【0021】以下、添付図面および請求の範囲を参照しながら実施例を詳細に説明し、本発明の様々な形態を明らかにする。

【0022】

【発明の実施の形態】本発明は代表的に、複数の入力源から入力を受け取れるワードプロセッサにおいて実施する。複数の入力源とは、非推測入力源および推測入力源を含む。キーボードおよびマウスによるテキスト入力は、非推測入力源の例である。非推測入力源の場合、コンピュータはユーザが意図したテキストを正確に決定できる。一方、推測入力源は、入力を推測結果に変換する。1つの推測結果は複数の候補を有し、各候補の正確さの確率は100%以下である。推測入力源の例は、音声認識装置である。音声認識装置は、音声入力をテキスト候補リストに変換する。コンピュータが音声を完璧に解釈することは不可能だからである。推測入力源の他の例は、手書き認識装置および入力システム（IME）で

ある。

【0023】ユーザは、ワードプロセッサを使ってテキストの一部を選択し、その選択テキストに対する候補を要求できる。選択テキストが1つ以上の推測入力源を元にコンピュータが作成したものであれば、その選択テキストには候補が存在する。ワードプロセッサは、選択テキスト用の候補を決定すると、グラフィカルユーザインターフェースを介してそれら候補を表示する。候補の1つをユーザが選択すると、ワードプロセッサはその選択候補で前記選択テキストを置き換える。

【0024】ワードプロセッサが提供する候補リストに希望の候補がなければ、ユーザはキーボードおよびマウスを使って前記選択テキストを編集する。例えばユーザは、前記選択テキスト中の1語を変更する。するとワードプロセッサは、その編集を取り入れて候補リストを修正し、修正した候補リストをユーザに提示する。修正した候補リストの1候補をユーザが選択すると、ワードプロセッサはその選択候補で前記選択テキストを置き換える。

【0025】推測入力結合器は、プログラムモジュールであり、選択テキストに対する候補を生成する。推測入力結合器はこれを行うため、選択テキストをテキスト要素に分解する。各テキスト要素は、1つの推測入力源に対応する。推測入力結合器は、各推測テキスト要素につき、その候補を代表する確率モデルを検索する。推測入力結合器は、検索した確率モデルと他のテキスト要素とを結合し、選択テキスト全体に対する候補リストを作成する。

【0026】推測入力結合器は、ワードプロセッサアプリケーションの一部であっても良いし、オペレーティングシステムの一部であっても良い。推測入力結合器は、オペレーティングシステムの一部ではなく、ワードプロセッサとインタフェースする独立したプログラムであっても良い。

【0027】ワードプロセッサは、選択テキスト用候補リストの向上を図るために、自然言語モデルを利用できる。自然言語モデルは、選択テキスト全体に自然言語原理を適用し、推測入力結合器が作成した各候補の適切さを再評価し、候補リストに新しい候補を追加する。自然言語モデルは、各テキスト要素に自然言語原理を適用することにより、候補リスト作成に使用する確率モデルの向上を図ることもできる。

【0028】テキスト要素は、連続した推測入力源を元にすることもある。少なくとも1つの推測入力源が推測結果を生成し、それを次の推測入力源への入力とする。この場合、第1推測入力源はユーザ入力を受け取り、最後の推測入力源はテキスト要素用の候補を生成する。推測入力結合器は、テキスト要素に対応する候補を含んだ連続確率モデルを取得する。この処理において次の推測入力源へ入力を与える際、推測入力結合器は前の入力源

の結果の選択をユーザに求めない。

【0029】推測入力結合器は、テキスト要素用の連続確率モデルを取得するため、選択テキストの元となったユーザ入力を選択し、そのユーザ入力を連続した推測入力源の第1推測入力源へ送る。次に推測入力結合器は、第1推測入力源が生成した各候補を第2推測入力源の入力とする。第2推測入力源は、受け取った候補の各々から推測結果を生成する。第2推測入力源からの推測結果は、総合候補リストにまとめる必要がある。さらに第3推測入力源が存在すれば、推測入力結合器は、総合候補リストの各候補を第3推測入力源への入力とする。連続した推測入力源における最後の推測入力源から作成する総合候補リストは、連続確率モデルである。

【0030】推測入力源のエラーは、複数の語にわたることが多い。ユーザは修正過程において、そのエラーの範囲全体に気付かないことがある。例えばユーザが「認識 (recognize、レコグナイズ)」と口述した時、音声認識エンジンが最適解釈として「素敵を破壊 (wreck a nice、レックアナイス)」と決定したとする。編集の時、ユーザは「破壊」を目にし、その語についてのみ候補を要求する。ユーザは、それに続く「素敵を」がやはり不正確であることに気付いていない。

【0031】不正確な隣接語を含めずにユーザがテキストを選択し、ワードプロセッサがその選択テキストのみを用いて候補リストを作成すれば、そのリストはユーザが意図したテキストに対応する候補を1つも持たないであろう。ユーザがリストから候補を選び、その選んだ候補で前記選択テキストを置き換えてしまうと、不正確な隣接語はそのままテキストに残る。

【0032】このような不都合を解消するため、推測入力結合器は、選択テキストを修正範囲モデルへ送る。修正範囲モデルは、修正範囲を拡大すべきか否かを決定する。前記「認識」の例では、修正用の適切な単位テキストは「素敵を破壊」であろう。この決定を下すに当たり、修正範囲モデルは、自然言語モデルと、類似エラーのモデルと、前記テキストをワードプロセッサへ提供した入力方法に関するモデルとが含む情報を参照する。入力方法に関するモデルとは、音声認識用音声モデル、手書き入力用手書きモデル、手話・身振り認識用視覚モデルを含む。

【0033】修正範囲モデルは、修正範囲を調整すべきと判断すると、1つ以上の単位テキストを特定する。この特定した単位テキストに対して、推測入力結合器は前記方法において候補を生成する。修正範囲モデルは、このような単位テキストのリストを推測入力結合器へ送る。推測入力結合器は、そのリストを処理する。

【0034】図面を参照しながら本発明の実施例を説明する。図中、同一番号は同一要素を示す。

【0035】動作環境例

図1を参照しながら本発明の実施に適したコンピュータ環境100の概略を説明する。一例としての動作環境100は、従来のパーソナルコンピュータシステム120を含む。このパーソナルコンピュータシステムは、処理装置121と、システムメモリ122と、システムバス123とを含む。システムバス123は、システムメモリ122を処理装置121に接続する。システムメモリ122は、リードオンリメモリ(ROM)124とランダムアクセスメモリ(RAM)125とを含む。ROM124は基本入出力システム126(BIOS)を含む。このBIOSは基本ルーチンを含む。この基本ルーチンは、起動時等にパーソナルコンピュータ120内の各要素間の情報転送を支援する。

【0036】パーソナルコンピュータシステム120は、ハードディスクドライブ127と、取出し可能磁気ディスク129を読み書きするための磁気ディスクドライブ128と、CD-ROM等の光媒体である取出し可能光ディスク131を読み書きするための光ディスクドライブ130とをさらに含む。ハードディスクドライブ127と磁気ディスクドライブ128と光ディスクドライブ130とは、各々ハードディスクドライブインターフェース132と磁気ディスクドライブインターフェース133と光ドライブインターフェース134とを介して、システムバス123に接続する。これらドライブおよび関連するコンピュータ読取り可能媒体は、不揮発性記憶手段をパーソナルコンピュータシステム120に提供する。ここに示したコンピュータ読取り可能媒体は、ハードディスク、取出し可能磁気ディスク、およびCD-ROMディスクだが、当業者には明らかなように、他のコンピュータ読取り可能媒体を、一例として示した前記動作環境に用いても良い。例えば磁気カセット、フラッシュメモリカード、デジタルビデオディスク、ベルヌイカートリッジ等を用いても良い。

【0037】ユーザは、キーボード140やマウス142等のポインティングデバイス等の入力装置を介して、パーソナルコンピュータシステム120にコマンドや情報を入力する。マイクロフォン161は、コンピュータシステム120に会話等の音声入力をを行うために使用する。ユーザは、コンピュータシステムに図や手書きのグラフィック情報を入力する場合、スタイルスを用いて筆記タブレット162にグラフィック情報を描く。コンピュータシステム120は、他の入力装置(図示せず)としてジョイスティック、ゲームパッド、衛星アンテナ、スキャナ、タッチスクリーン、スタイルス等を含んでも良い。マイクロフォン161は、システムバスに接続したオーディオアダプタ160を介して処理装置121に接続できる。他の入力装置は、システムバスに接続したシリアルポートインターフェース146を介して処理装置121に接続する場合が多い。しかしながらゲームポート、ユニバーサルシリアルバス(USB)等のインターフ

ェースによって接続することもできる。

【0038】モニタ147等の表示装置は、ビデオアダプタ148等のインターフェースを介してシステムバス123に接続する。モニタに加え、パーソナルコンピュータシステムは、スピーカやプリンタ等の周辺出力装置(図示せず)を一般に含む。

【0039】パーソナルコンピュータシステム120は、ネットワーク環境で動作させ、1つ以上の遠隔コンピュータ149に論理接続しても良い。遠隔コンピュータ149は、サーバ、ルータ、ピア装置、共通ネットワークノード等であり、一般に前記パーソナルコンピュータシステム120に関連して説明した要素の多くあるいは全てを含む。ただし図1は記憶装置150のみを示す。図1に示す論理接続は、ローカルエリアネットワーク(LAN)151と、ワイドエリアネットワーク(WAN)152とを含む。このようなネットワーク環境は、事務所、企業コンピュータネットワーク、インターネット、インターネットにおいて普及している。

【0040】LAN環境で使用する場合、パーソナルコンピュータシステム120は、ネットワークインターフェース153を介してローカルネットワーク151に接続する。WAN環境で使用する場合、パーソナルコンピュータシステム120は、一般にモデム154等の手段を含み、インターネット等のワイドエリアネットワーク152を介して通信を確立する。モデム154は内蔵または外付けであり、シリアルポートインターフェース146を介してシステムバス123に接続する。ネットワーク環境の場合、パーソナルコンピュータシステム120のプログラムモジュールは、遠隔記憶装置150に格納しても良い。図示のネットワーク接続は例であり、コンピュータシステム間の通信リンクは他の手段で確立しても良い。また本発明は、パーソナルコンピュータシステム以外に、ホストまたはサーバコンピュータシステムでも実現可能であり、CD-ROM以外の手段、例えばネットワーク接続インターフェース153によってホストコンピュータシステムにデータを送ることができる。

【0041】多くのプログラムモジュールをコンピュータシステム120の駆動装置やRAM125に格納できる。プログラムモジュールは、コンピュータシステム120の機能を制御し、ユーザ、入出力装置、あるいは他のコンピュータと対話する。プログラムモジュールは、ルーチン、オペレーティングシステム135、アプリケーションプログラムモジュール138、データ構造、ブラウザ、他のソフトウェア部品、あるいはファームウェア部品を含む。本発明は、推測入力結合プログラムモジュール137、推測入力インターフェースプログラムモジュール139等の1つ以上のプログラムモジュールによって好適に実施できる。これらモジュールの各々は、発明の詳細な説明に記載した方法に基づく。

【0042】アプリケーションプログラムモジュール1

38は、本発明に関連して使用する様々なアプリケーションからなる。図2はその幾つかを示す。これらプログラムモジュールの目的および相互作用は、図2を参照して詳細に説明する。これらプログラムモジュールは、ワードプロセッサプログラム210（ワシントン州レドモンドのマイクロソフト社の製品であるワード等）、手書き認識プログラムモジュール230、音声認識プログラムモジュール240、および入力システム（IME）250を含む。

【0043】以下に説明する様々な手順を実行するための特定のプログラミング言語は説明しない。理由は、以下の説明および添付図面における動作、処理段階、および手順は、十分な開示を行っているため、当業者であればそれらに基づいて本発明の実施例を実現可能だからである。さらに実施例の実行に使用可能なコンピュータおよびオペレーティングシステムは多数あり、これら多くの異なるシステムに適用可能な詳細コンピュータプログラムを提示するのは困難だからである。特定コンピュータのユーザは、そのユーザの必要性および目的に最も適した言語およびツールを知っているであろう。

【0044】当業者には明らかなように、本発明は他のコンピュータシステム構成でも実施可能である。例えば携帯装置、マルチプロセッサシステム、マイクロプロセッサベースの消費者用電子機器、プログラム可能消費者用電子機器、マイクロコンピュータ、メインフレームコンピュータ等でも実施可能である。本発明は分散コンピュータ環境でも実施可能である。分散コンピュータ環境では、通信ネットワークでリンクした各遠隔処理装置によってタスクを実行し、プログラムモジュールを局所記憶装置および遠隔記憶装置の両方に置くことができる。

【0045】プログラムモジュールの概要
図2は、複数入力源データ処理システム200のプログラムモジュールの全体を示す。これらプログラムモジュールの主たる目的は、ユーザが推測入力源および非推測入力源を用いて、ワードプロセッサ等のアプリケーション210にテキストを入力できることである。代表的な推測入力源は、手書き認識プログラムモジュール230、音声認識プログラムモジュール240、入力システム（IME）250、音声認識プログラムモジュール260である。キーボード140は、代表的な非推測入力源である。ユーザは、1つ以上のこれら入力源を介してワードプロセッサ210にテキストを入力し、テキストの一部を選択し、その選択テキストに対する候補リストを要求できる。この選択テキストは、複数の推測入力源および非推測入力源からの入力を含むことがある。選択テキストが少なくとも1つの推測入力源からのデータを含んでいれば、その選択テキストに対する候補が存在する。プログラムモジュールは、その候補リストを作成し、グラフィカルユーザインタフェースを通してそれをユーザに提示する。ユーザが候補の1つを選

ぶと、選んだ候補で前記選択テキストを置換する。推測入力源230、240、250、260の動作を順次説明する。

【0046】手書き認識プログラムモジュール230は、ユーザからの手書き入力280を受け取る。ユーザは、スタイルスを用いて筆記タブレット162に筆記することにより、手書き入力280を生成できる。他の装置を用いても良い。例えばマウス142を用いて、モニタ147に書いても良い。スタイルスを用いて、タッチスクリーンに書くこともできる。手書き入力280は、好ましくはオペレーティングシステム135の筆記タブレットドライバモジュールにより、手書き認識プログラムモジュール230へ送る。

【0047】コンピュータで手書きを解釈することは難しい。手書き認識プログラムモジュール230は、手書き入力280を完全に解釈できるとは限らない。プログラムモジュール230が行う最良のことは、手書き入力280の候補を生成し、各候補に正当性の確率を割り当て、推測結果を提供することである。確率モデル270aは、データ構造を含む。このデータ構造は、手書き認識プログラムモジュール230が手書き入力280を処理して生成した確率データを含む。

【0048】確率モデル270が確率データを格納するために使用するデータ構造は、どのようなものでも構わない。2つの有用なデータ構造は、格子およびベストn候補リストである。格子は、当業者に良く知られた構造であるため、詳細説明は省略する。簡単に説明すると、格子は、推測入力源が生成した語または句をノードに格納する。各語または句は確率データであるため、各ノードは対応する語または句に割り当てた確率も格納する。当業者に公知の方法を用いて格子を探索し、各テキスト要素用の候補を生成できる。テキストの隣接部を代表する複数の格子を連結してメタ格子を作成できる。このメタ格子を探索することにより、隣接するテキスト要素の候補を生成できる。

【0049】確率データは、ベストn候補とそれらに割り当てた確率とのリストでも良い。ある語または句に対するベストn候補リストは、その語または句を代表する格子から作ることもできる。

【0050】音声認識プログラムモジュール240は、手書き認識プログラムモジュール230と同様に動作する。ただし音声認識プログラムモジュール240は、マイクロフォン161を介してユーザから音声入力290を受け取る。マイクロフォン161は、オペレーティングシステム135のマイクロフォンドライバモジュールで駆動する。音声は解釈が難しい。発音が似て意味やスペルが異なる語がたくさんあるからである。従って音声認識プログラムモジュール240も、推測結果を生成する。確率モデル270bは、データ構造を格納する。このデータ構造は、音声認識プログラムモジュール240

が音声入力290を処理して生成した確率データを含む。

【0051】入力システム(IME)250も、確率データを生成する。IME250は、入力を外國語テキストに変換する。IME250への入力は、例えばキーボード140およびマウス142を介してコンピュータへタイプ入力したテキストである。確率モデル270cは、データ構造を含む。このデータ構造は、IME250が生成した確率データを含む。

【0052】IME250は、アジア言語等において表意文字を生成する時、特に有用である。かかる言語における表意文字数は、キーボードのキー数よりはるかに多いからである。IME250を使わなければ、特定の表意文字をコンピュータに入力するのは難しい。代表的なIME250の場合、ユーザは所望の中国語文字の音声スペルを英字でタイプする。多くの中国語文字は発音が類似しているため、タイプした音声スペルは、多くの異なる中国語文字の1つを意図している。そこでIME250は、推測結果を生成すると共に、タイプした音声スペルが意図したであろう最も確率の高い候補をユーザに提示する。ユーザは、その候補から正しい1つを選択できる。

【0053】1つの推測入力源が生成する推測結果は、次の推測入力源への推測入力とすることもできる。この場合、これら推測入力源は「連続した推測入力源」であり、連続配置していると考えられる。これはプログラムモジュールの構成293に示すとおりである。この構成293は、IME250の他の実施例でもある。

【0054】この実施例において、英語音声入力262をコンピュータに入力し、それを使って日本語テキストを生成できる。音声262は、音声認識プログラムモジュール260へ送る。音声認識プログラムモジュール260の動作は、音声認識プログラムモジュール240の動作とほぼ同じである。モジュール260を独立して示した理由は、それが異なる音声解釈エンジンを持てるからである。例えば音声認識プログラムモジュール260は、音声認識プログラムモジュール240とは異なる言語を解釈できる。確率モデル270dは、データ構造を含む。このデータ構造は、音声認識プログラムモジュール260が音声入力を処理して生成した確率データを含む。

【0055】英語音声・日本語IMEの場合、音声認識プログラムモジュール260は、英語音声から英語テキスト候補を生成し、それを確率モデル270dに格納できる。確率モデル270dに格納した1つ以上の英語テキスト候補は、IME250への入力となる。IME250は、その英語テキスト入力を日本語文字に変換する。IME250への各候補入力は、個別の推測結果を生成する。ただしIME250への2つの個別入力に基づく推測結果間には、候補の重複があり得る。

【0056】図2において、音声認識プログラムモジュール260からIME250への矢印は、音声認識プログラムモジュール260がIME250の推測入力源であることを示している。しかしこれら2つのプログラムモジュールは、直接インターフェースしているわけではない。音声認識プログラムモジュール260からIME250への推測入力は、例えば推測入力インターフェース139等のインターフェースプログラムモジュールを中継する。この推測入力インターフェース139に対して、各推測入力源は直接接続している。

【0057】推測入力インターフェース139は、推測入力源と、アプリケーション210との間で、推測データを受け渡す。ここで推測入力源は、例えば手書き認識プログラムモジュール230、音声認識プログラムモジュール240、IME250である。推測データ通路として推測入力インターフェース139を設けと、推測データを受け取るアプリケーション210と推測入力源との通信が簡略になる利点がある。すなわちアプリケーションは、推測入力インターフェースとの通信方法を持つだけ良く、推測入力源に対しての通信方法を持たないで良い。推測データを受け取るアプリケーション210は、本発明の一実施例においてワードプロセッサである。アプリケーション210は、スプレッドシート、グラフ、電子メールプログラム、音楽編集プログラム、CADプログラム、プレゼンテーションソフトウェア(ワシントン州レドモンドのマイクロソフト社の製品であるパワーポイント等)、オペレーティングシステム等のソフトウェアプログラムでも良い。

【0058】ワードプロセッサ実施例の場合、ワードプロセッサ210は、推測入力インターフェース139を介して、最適候補テキストを受け取る。各最適候補は、ワードプロセッサにデータを入力した各推測入力源からのものである。複数の推測入力源からワードプロセッサ210にデータを送ることに加え、例えばキーボード140でタイプすることにより、非推測データをワードプロセッサに入力できる。ワードプロセッサ210は、これら元データを結合して複数入力源テキストを作成し、それをユーザに提示する。ワードプロセッサ210は、そのテキストの各要素の入力源をユーザに示さないが、記録している。

【0059】ワードプロセッサ210は、ユーザがテキストの一部を選択し、その選択テキストに対する候補を要求することを可能にする。選択テキストが1つ以上の推測入力源を元にしていれば、候補が存在する。ワードプロセッサ210は、選択テキスト自体とその選択テキストの各要素の入力源とを推測入力インターフェース139へ送り、候補リストを要求する。推測入力インターフェース139は、その要求を処理し、全選択テキストに対する候補リストをワードプロセッサ210に返す。ワードプロセッサは、グラフィカルユーザインターフェースを

介してその候補リストをユーザに提示する。候補リストからユーザが1つを選ぶと、ワードプロセッサは、選んだ候補で選択テキストを置き換える。

【0060】候補リスト要求を処理する際、推測入力インターフェース139は、その要求を推測入力結合器137へ送る。推測入力結合器137は、推測入力インターフェース139を介して推測入力源と通信し、確率モデル270に関する必要情報を検索し、候補リストを作成する。

【0061】推測入力結合器137は、候補リスト作成の際、自然言語モデル220を参照しても良い。この場合、結合器137は、確率モデル270から検索した情報を用いて、中間候補リストを作成し、それを自然言語モデル220へ送る。自然言語モデルは、文法、テキスト全体の意味、様々な語順の確率等を手掛かりに、中間候補リストを分析する。これに基づき自然言語モデル220は、追加候補を生成し、中間候補リスト中の候補の確率を再評価する。選択テキストの候補リストを生成する方法を図3～9および11を参考しながら説明する。

【0062】図2において、推測入力源230, 240, 250は、各自の推測データを他の推測入力源を介して選別することなく、ワードプロセッサ210に提供できる。すなわち推測入力源230, 240, 250は、各々が直接（ただし推測入力インターフェース139を介して）推測データをワードプロセッサ210へ送ることができる。各入力源からの推測データは、ワードプロセッサの同一文書に含めることができる。従ってこれら入力源は、並列推測入力源296であり、並列構成と言える。

【0063】これまで各プログラムモジュールは別々に説明してきたが、当業者には明らかなように、これらプログラムモジュールは、様々に組み合わせることが可能であり、新しいプログラムモジュールを作成して同様の結果を得ることもできる。特に推測入力結合器137と自然言語モデル220とは、推測入力インターフェース139に含めることができる。これら3つのプログラムモジュールは、オペレーティングシステム135またはワードプロセッサ210の一部とすることもできる。結合器137および自然言語モデル220は、各独立したプログラムとし、各々がワードプロセッサ210と直接インターフェースしても良い。また推測入力源230, 240, 250, 260は、各々が独立アプリケーションプログラムモジュール138でも良く、オペレーティングシステム136の一部でも良い。

【0064】代表的実施例の図示図3, 5, 11は、本発明実施例においてユーザが目で見ながら行う状況を示すと共に、推測入力結合器137と自然言語モデル220との機能を示す。

【0065】図3において、コンピュータ120は、複数のテキスト入力方法を有し、ユーザからの入力を受け

取り、それをワードプロセッサ等のアプリケーション210へ送る。コンピュータは、その入力をテキスト300に変換し、モニタ147に表示する。本例において、ユーザは「THIS IS A MESSAGE WRITTEN BY A THOUSAND MONKEYS TYPING AT RANDOM.（これは千匹の猿がでたらめにタイプして書いたメッセージです。）」とのテキストを意図した。ところがコンピュータは、それを「THIS IS A MESSAGE WRITTEN BY A TOWN OF MY KEYS TAIPING AT RANDOM.（これはでたらめに私の鍵のタイピンの町が書いたメッセージです。）」と解釈し、テキスト300を生成した。

【0066】テキスト300が表示されると、ユーザはその一部を強調表示することにより、選択テキスト310を作成する。図3の選択テキスト310は、3つのテキスト要素312, 314, 316を有する。これらテキスト要素は、各々が異なる推測入力源から生成されている。例えばテキスト要素312は、手書き認識プログラムモジュール230が手書き入力280を処理して生成した候補の1つである。この推測入力源から生成した候補は、確率モデル270aが保持する。候補リスト318は、確率モデル270aが格納するテキスト要素312用候補リストである。この候補リストからコンピュータは、「TOWN OF（の町）」を選んだ。その理由は、テキスト要素312の元となつた手書き入力280に対し、候補リスト318の「TOWN OF（の町）」が最適であるとコンピュータが判断したからである。しかしながら候補リスト318が示すように、コンピュータは「GO WNOF（のガウン）」および「THOUSAND（千匹の）」もテキスト要素312の候補として認識している。

【0067】音声認識プログラムモジュール240は、音声入力290を処理して推測結果を生成し、それを確率モデル270bに格納する。候補リスト320は、テキスト要素314に対して確率モデル270bが格納している候補である。ここでは「MY KEYS（私の鍵）」が最適であるとして選択された。

【0068】テキスト要素316「TAIPING（タイピン）」は、第3の推測入力源から来ている。この推測入力源の候補は、確率モデル270cに格納されており、候補リスト322に含まれる。

【0069】推測入力結合器137は、候補リスト318, 320, 322から様々な組合せを作成し、各組合せの順位リストを作成する。この順位付けは、選択テキスト310に対する各組合せの正当性の確率を計算して行う。選択テキスト310に対する候補のうち上位を、候補リスト330としてモニタ147に表示する。

【0070】図4において、候補リスト330の表示後、ユーザは選択テキストを編集することもできる。図4においてユーザは、タイピング編集410により、「TOWN OF（の町）」を「THOUSAND（千匹の）」で上書きした。この結果、選択テキスト310

のテキスト要素312は、テキスト要素312'に置換された。

【0071】編集410は、キーボード140とマウス142を用いて行うこともできる。このような入力方法は、非推測入力であるため、テキスト要素312'に対する候補は無い。この変更は候補リスト318'に反映する。候補リスト318'は、図3の候補リスト318に置き換わる。候補リスト318'は、「THOUSAND(千匹の)」が唯一の候補である。

【0072】編集が完了すると、推測入力結合器137は、候補リスト318'、320、322から再び様々な組合せを作り、編集後選択テキスト310用の候補を作成する。これら候補は、候補リスト430としてモニタ147に表示する。

【0073】選択テキスト310を置換する候補として、候補リスト430に表示されているもの以外をユーザが希望する場合、ユーザは再び選択テキスト310を編集する。これを図11に示す。ユーザは、図4の選択テキスト310にある「MYKEYS(マイ キーズ：私の鍵)」に「MONKEYS(モンキーズ：猿)」を上書きタイプして編集412を行う。その結果、テキスト要素314' 「MONKEYS(猿)」がテキスト要素314 「MY KEYS(私の鍵)」に置き換わる。そして候補リスト320は候補リスト320'に置き換わる。この候補リスト320'は、1つの候補しか含まない。なぜなら前記編集が非推測的だからである。推測入力結合器137は、再び各候補リスト318'、320'、322の候補を組合せ、選択テキスト310に対する候補リスト430'を作成する。

【0074】この時、ユーザは選択テキスト310用の候補として、適切な候補414を見つけ、その候補414を強調表示させて選択する。すると候補414は選択テキスト310に置き換わり、新しいテキスト416を生成する。

【0075】図5は図3と同様であるが、自然言語モデル220を動作させた場合の本発明実施例を示す。図3に示したように、ユーザはテキスト要素312、314、316からなる選択テキスト310を選ぶ。推測入力結合器137は、これらテキスト要素の様々な候補組合せを作成し、選択テキスト310用中間候補リストを作成する。結合器137は、最適候補を表示することに代えて、前記中間候補リストを自然言語モデル220へ送る。

【0076】自然言語モデル220は、選択テキスト310全体に自然言語原理を適用することにより、中間候補リストの各候補の確率を再評価する。この再評価は、選択テキストが含む文法等の言語的手掛かりの分析を含む。自然言語モデル220は、中間候補リストが含まない候補を追加することもある。自然言語モデル220は、中間候補リストの再評価および生成した追加候補に

基づいて修正候補リストを作成し、それを推測入力結合器137に返す。結合器137は、修正候補リストから上位候補を選択し、候補リスト530として表示する。【0077】自然言語モデルの動作により、候補リスト530は、図3の候補リスト330より優れていることが多い。従ってユーザは、候補リスト530から候補502を選択でき、選択テキスト310を編集する必要がない。図5の例においてユーザは、選択テキスト310を候補502と置き換えることにより新しいテキスト504を作成する。

【0078】代表的実施例のフローチャート

図6は、元データを処理するための代表的ルーチン600の各ステップを示すフローチャートである。このルーチンは、図3、5、11に示した実施例を実行するための各段階を含む。ルーチン600は、ステップ602において開始する。ワードプロセッサは、複数の入力源から元データを受け取る。入力源は、手書き認識プログラムモジュール230、音声認識プログラムモジュール240、入力システム250等の推測入力源でも良い。あるいはキーボード140およびマウス142を使ってタイピング入力した非推測入力でも良い。あるいは連続した2つ以上の推測入力源からのデータでも良い。連続した推測入力源からのデータの場合、各推測入力源は、それぞれ異なる入力源として扱うことができる。

【0079】ワードプロセッサは、複数の入力源から元データを受け取ると、ステップ604においてそのデータを複数入力源テキスト列に結合する。すなわち、ワードプロセッサは、元データに対応するテキストを生成してモニタ147に表示可能とする。さらにワードプロセッサは、そのテキストの各語の入力源を記録するためのデータ構造を生成する。

【0080】ステップ606において、ユーザは必要に応じて表示テキストの一部を選択する。この選択テキストは、複数の入力源を有するテキストでも良い。例えばユーザは、選択テキストの開始位置でマウスボタンを押し、選択テキストの終端までマウスをドラッグし、そこでマウスボタンを離す。ワードプロセッサは、その選択テキストを強調表示し、該当部分を明示することが好ましい。

【0081】ステップ608においてワードプロセッサは、選択テキストに対する「候補表示」コマンドを受け取る。それに応じてワードプロセッサは、ステップ610において、複数の入力源を有する選択テキスト用の候補を決定する。

【0082】ステップ612においてワードプロセッサは、それら候補をモニタ147に表示する。ワードプロセッサは、グラフィカルユーザインタフェースを介して、それら候補を確率順に表示することが好ましい。このグラフィカルユーザインタフェースは、ユーザが表示候補の1つを選択することを可能にする。グラフィカル

ユーザインタフェースは、サブウインドウに表示しても良い。ユーザはそのサブウインドウを任意に移動することにより、その下に隠れたテキストを表示できる。

【0083】ステップ614においてユーザは、ワードプロセッサにコマンドを与える。このコマンドは例えば、表示候補の選択、選択テキストの編集、あるいは選択テキスト外のテキストの一部にマウスを移動しボタンを押すことによる新たなテキスト選択等である。

【0084】ステップ616において、ワードプロセッサは、ユーザが表示候補の1つを選択したか否かを決定する。ユーザが表示候補の1つを選択していれば、ステップ618においてワードプロセッサは、その選択候補で選択テキストを置換する。次にステップ624においてワードプロセッサは、候補の表示を中断する。そしてステップ626においてルーチンは終了する。ステップ626の後、ステップ602に戻ってルーチンを繰り返すこともできる。

【0085】ステップ616においてユーザが表示候補を選択しなければ、ステップ620を実行する。ステップ620においてワードプロセッサは、ユーザが選択テキスト内のテキストを編集したか否かを決定する。ユーザが選択テキスト内のテキストを編集していれば、ステップ622においてワードプロセッサは、その編集を処理する。ワードプロセッサが編集処理を完了すると、ルーチンはステップ610にループし、編集後選択テキスト用の新しい候補を決定する。

【0086】ステップ620において、ステップ614で受け取ったユーザコマンドが選択テキストの編集コマンドでなければ、ワードプロセッサはステップ624を実行する。この場合ユーザは、選択テキストの外側でマウスを押すことにより、新しい選択テキストの生成を開始している。従ってワードプロセッサは、ステップ624において候補の表示を中断し、ステップ626においてルーチンを終了する。ここで再びステップ602に戻ってルーチンを繰り返すこともできる。

【0087】図7は、図6のルーチン610の詳細を示す。このルーチンは、複数入力源を有する選択テキストに対して、候補を決定するための各ステップを含み、例えば推測入力結合器が実行する。結合器は、ワードプロセッサ内のプログラムモジュール、オペレーティングシステム内の独立プログラム、またはワードプロセッサとインタフェースする独立プログラムである。

【0088】ルーチンはステップ702で開始する。ここで推測入力結合器は、選択テキストを解析し、入力源別のテキスト要素に分解する。すなわち推測入力結合器は、モニタ147に表示したテキストの各語の入力源を格納したデータ構造を参照する。選択テキストをテキスト要素に分解することにより、選択テキスト用候補の決定が管理しやすくなる。

【0089】当業者には明らかなように、テキスト要素

の定義は多様である。図7～9に示したものとは異なる定義を用いても良い。この場合、解析ステップ702や他のルーチンを適切に変更する必要がある。テキスト要素は、例えば単一の語、または同一入力源を有する複数の語からなる句である。推測入力から得た句の場合、その句の中間に編集を挿入すると、元の句と編集とは異なるテキスト要素となり得る。

【0090】図7および図9において、テキスト要素は、隣接のテキスト要素とは異なる1つまたは連続した推測入力源を有する最大単位テキストと定義し、このテキスト要素は、タイピングにより編集したテキストを包むことができる。タイピングにより作成した単位テキストであって、推測入力源からのテキスト要素の編集ではない場合、その単位テキストはテキスト要素と見なす。例えば異なる推測入力源を元とする2つのテキスト要素の間に、ユーザがタイピングによってテキストを挿入した場合、この挿入したテキストはそれ自体がテキスト要素である。

【0091】ステップ704において推測入力結合器は、独自の推測入力源を有するテキスト要素について、対応する確率モデルを検索する。ステップ706において推測入力結合器は、確率モデルが全て格子であるか否かを決定する。全ての確率モデルが格子であれば、ステップ708を実行する。

【0092】ステップ708において推測入力結合器は、検索した全格子を連結してメタ格子を作成する。このメタ格子を作成するに当たり、推測入力結合器は、タイプしたテキスト要素用ノードを生成する。タイプしたテキスト要素は、推測入力源を元とするテキスト要素に含まれない。ステップ710において入力結合器は、当業者に公知の格子探索方法を用いてメタ格子を探査し、選択テキスト用候補リストを作成する。次にステップ716に移る。このステップは、ステップ706の「NO」の分岐の後に説明する。

【0093】ステップ706に戻り、ステップ704で検索した全確率モデルが格子でなければ、推測入力結合器はステップ712を実行する。この場合、少なくとも1つの確率モデルがベストn候補リストである。従って推測入力結合器は、各格子をベストn候補リストに変換し、ステップ714を実行する。

【0094】ステップ714において推測入力結合器は、ベストn候補リストとタイプしたテキスト要素とを結合し、結合候補リストを作成する。これは、ベストn候補のあらゆる組合せを形成することによって行う。推測入力結合器は、各組合せについて、選択テキストに現れる順序に候補を配置する。そして選択テキスト用候補リストを形成する。ステップ714からステップ716へ進む。

【0095】ステップ710または714からステップ716へ進む時、推測入力結合器は、選択テキスト用候

補リストを作成済みである。ステップ716はオプションであり、自然言語モデルを利用できる。このステップにおいて推測入力結合器は、選択テキスト用候補リストを自然言語モデルへ送る。

【0096】ステップ716を実行する時は、ステップ718も実行する。このステップにおいて、自然言語モデルは、順位付けした修正候補リストを推測入力結合器へ返す。この修正候補リストは、自然言語モデルへ送った候補リストの各候補の確率を再評価した結果を含む。修正候補リストはさらに、選択テキストに対して自然言語モデルが生成した新しい候補も含む。自然言語モデルは、自然言語原理を用いて修正候補リストを生成する。自然言語原理は、自然言語モデルが選択テキスト全体に適用する文法等の言語的手掛かりの分析を含む。

【0097】ステップ720において推測入力結合器は、表示用に上位候補を選択する。オプションのステップ716および718を実行した場合、推測入力結合器は、自然言語モデルから受け取る修正候補リストから表示用候補を選択する。ステップ716および718を実行しない場合、推測入力結合器は、ステップ710または714において作成した候補リストから表示用候補を選択する。ステップ720の後、ステップ722においてルーチンは終了する。

【0098】図8は、図7のルーチン704の各ステップを示す。このルーチンにおいて推測入力結合器は、1つの推測入力源を元とする各テキスト要素に関し、確率モデルを検索する。ステップ802においてルーチンを開始し、各テキスト要素を処理するループに入る。ステップ802において推測入力結合器は、テキスト要素を検索する。ステップ804において推測入力結合器は、当該テキスト要素が推測テキスト要素であるか否かを決定する。推測テキスト要素でなければ、ステップ806へ進む。この場合、当該テキスト要素は、例えばキーボードとマウスを用いてタイプしたテキストである。このテキスト要素は非推測であるため、推測入力結合器は、そのテキスト要素に100%の確率を割り当てる。次に推測入力結合器は、ステップ818を実行する。このステップは後述する。

【0099】ステップ804に戻り、ステップ802で検索したテキスト要素が推測テキスト要素であれば、推測入力結合器はステップ808を実行する。このステップにおいて推測入力結合器は、そのテキスト要素が連続した推測入力源からか否かを決定する。連続した推測入力源からであれば、推測入力結合器はステップ810を実行し、連続した推測入力源の最後の推測入力源が生成した連続確率モデルを取得する。ルーチン810の後、推測入力結合器はステップ812を実行する。ステップ808において、ステップ802で検索したテキスト要素が連続した推測入力源からでない場合も、ステップ812へ進む。

【0100】ステップ812において推測入力結合器は、ユーザがキーボードおよびマウスを使用してテキスト要素を編集したか否かを決定する。編集していれば、ステップ814において推測入力結合器は、対応する確率モデルを更新する。この更新は、対応する確率モデルが格子であれば、当該テキスト要素から削除された語に対応するノードの削除を含む。また推測入力結合器は、当該テキスト要素に新しい語があれば、それに対応してノードを追加する。確率モデルがベストn候補リストであれば、推測入力結合器は、削除された語を削除し、挿入された語を追加することにより、リストの各候補を更新する。

【0101】ステップ814からステップ816へ進む。ステップ812においてユーザが当該テキスト要素を編集していないと決定した場合も、ステップ816へ進む。ステップ816において、推測入力結合器は、ステップ802で選択したテキスト要素に対し、確率モデル結果を検索する。当該テキスト要素が連続した確率モデルから得たものであれば、検索した確率モデルはステップ810で作成した連続確率モデルである。また当該テキスト要素が編集されていれば、ステップ814で更新した連続確率モデルである。検索した確率モデルは、格子またはベストn候補リストである。検索した確率モデルは、選択テキスト要素に関する情報を含んでいれば良いので、推測入力結合器は、当該テキスト要素を含む大きなテキスト用確率モデルから当該確率モデルを検索することもできる。

【0102】ステップ802で選択したテキスト要素は、推測入力からのものでも、候補を代表する確率モデルが用意されていない場合がある。この場合、当該テキスト要素は非推測テキスト要素と同様に扱う。すなわち推測入力結合器は、当該テキスト要素用の既知の候補に100%の確率を割り当てる。推測入力結合器は、ステップ816の次にステップ818を実行する。

【0103】ステップ818は、ステップ816または806に統合して実行する。ステップ818において推測入力結合器は、選択テキスト内に未処理のテキスト要素があるか否かを決定する。未処理のテキスト要素があれば、ルーチンはステップ802にループし、推測入力結合器は次のテキスト要素を取得し処理する。

【0104】ステップ818において未処理のテキスト要素が無ければ、推測入力結合器はオプションのステップ820を実行し、自然言語モデルを利用する。このステップにおいて推測入力結合器は、テキスト要素用に検索した各確率モデルを自然言語モデルへ送る。自然言語モデルは、そのテキスト要素に自然言語原理を適用し、結果を推測入力結合器に返す。ステップ820において自然言語モデルは、選択テキスト全体ではなく各テキスト要素に対して動作するため、ステップ820は、ステップ716および718の代替としてあるいは追加とし

て実行する。ステップ820の後、ルーチンはステップ822で終了する。

【0105】図9は、図8のルーチン810の各ステップを示す。このルーチンにおいて推測入力結合器は、連続した推測入力源を持つテキスト要素に対応する連続確率モデルを得る。

【0106】ルーチン810はステップ902で開始する。このステップは、連続した推測入力源のうち、最後の推測入力源を除く全てを順次処理するためのループを開始する。第1回目のステップ902において、推測入力結合器は、連続した推測入力源から第1推測入力源を選択する。この第1推測入力源は、当該テキスト要素を生成したユーザ入力を受け取った入力源である。

【0107】第1推測入力源は複数の候補を生成し、これらは第2推測入力源への入力となる。第2回目以降のステップ902では、推測入力結合器は、ステップ914において（後述する）、ステップ902で選択した推測入力源用の総合候補リストを作成している。この総合候補リストは、選択した推測入力源に関連する候補を含んでおり、これら候補は次の推測入力源の入力として使われる。ステップ904は、選択した推測入力源に関する全候補を処理するループを開始する。ステップ904において推測入力結合器は、選択した入力源用候補の1つを取り出す。

【0108】ステップ906において推測入力結合器は、ステップ904で取り出した候補を次の推測入力源へ入力として渡す。次の入力源もやはり推測入力源であるため、同様に推測結果を生成する。推測入力結合器は、この推測結果を取り出す。

【0109】ステップ908において推測入力結合器は、ステップ906で取り出した推測結果が格子であるか否かを決定する。格子でなければ順位付け候補リストであるため、ステップ912（後述する）へ進む。格子であれば、ステップ910において推測入力結合器は、その格子を順位付け候補リストに変換し、各候補に確率を割り当てる。次にステップ912へ進む。

【0110】ステップ912において、推測入力結合器は、選択した入力源はさらに候補を持つか否かを決定する。さらに候補があれば、ルーチンはステップ904に戻り、推測入力結合器は次の候補を取得する。さらに候補がない場合は、ステップ914へ進む。

【0111】ステップ914において推測入力結合器は、ステップ902で選択した入力源用候補を後続の推測入力源への入力として生成した候補リストを全て結合し、総合候補リストを作成する。この総合候補リストを作成するに当たり、推測入力結合器は、各候補リストの各一意の候補に対して1つの項目を総合候補リストに準備する。そして推測入力結合器は、総合候補リスト中の各候補の確率を計算する。これは、各候補リストにおける各候補の確率を合計して行う。推測入力結合器は、各

合計した確率を総合候補リスト中の対応する候補に割り当てる。

【0112】ステップ916において、推測入力結合器は、現在次の入力源となっている入力源の後にさらに別の入力源が存在するか否かを決定する。さらに別の入力源が存在すれば、推測入力結合器は、ステップ902において現在次の入力源となっている入力源を選択し、前記さらに別の入力源を次の入力源とする。この時、推測入力結合器は、前記総合候補リストの各候補を選択した入力源への入力とする。

【0113】ステップ916においてさらに別の入力源がなければ、ルーチンはステップ918において終了する。ステップ914で作成した最新の総合候補リストは、連続確率モデルとなる。

【0114】図10は、元データ処理ルーチンの別の実施例1000における各ステップを示す論理フローチャートである。このルーチンは、ユーザのテキスト選択に応じて修正する単位テキストを自動的に調整するルーチンである。

【0115】ルーチン1000はステップ1005で開始する。このステップにおいて、ワードプロセッサ210は、ユーザが修正するために選択した選択テキストを受け取る。ユーザは、マウス142を用いて語または語群を選択することにより前記選択テキストを指定する。あるいはユーザは、マウスを用いて单一の語からなる選択テキストを当該語の上または近くにマウスポインタをおくことにより指定する。ワードプロセッサ210は、その選択テキストを推測入力結合器137へ送り、修正候補を決定する。

【0116】ステップ1010において推測入力結合器137は、選択テキストを修正範囲モデルへ送り、修正範囲を調整すべきか否かを決定する。修正範囲の調整は例えば、選択テキストよりも良いテキスト修正候補をユーザに提供できる単位テキストを特定することを含む。例えば選択テキストは、エラーを含んだ隣接語を包含していないことがある。このような隣接語のエラーは、選択テキストと共に修正すべきものである。一般に、選択テキストの隣接語が含むエラーは、選択テキスト内のエラーに関連しているため、特定可能である。

【0117】従って修正範囲モデルが指定する単位テキストは、選択テキストに1つ以上の隣接語を追加したものとなる。修正範囲モデルは、修正すべきテキストとして、単一の単位テキストではなく、複数の単位テキストを指定することもできる。この場合、各単位テキストは、テキスト修正用として優れた候補を提供する可能性がある。

【0118】ステップ1015において、推測入力結合器137は、ユーザに提供すべき修正候補用の単位テキストリストを修正範囲モデルから受け取る。修正範囲モデルが、修正範囲の調整は必要ないと判断すれば、当該

単位テキストリストは選択テキストだけを含む。修正範囲モデルが、修正用として1つの単位テキストのみを指定すれば、前記単位テキストリストはその1つの単位テキストのみを含む。ステップ1020は、単位テキストリスト中の各単位テキストを処理するためのループを開始する。この単位テキストリストは、ステップ1015において修正範囲モデルが推測入力結合器137に返したものである。ステップ1020において結合器137は、処理する単位テキストを選択する。ステップ1025において結合器137は、図7のルーチンの各ステップを実行し、選択した単位テキスト用の候補を決定する。結合器137が図7の各ステップを実行することによってルーチン1025を行う際、図7の「選択テキスト」は、選択した単位テキストに対応する。

【0119】ステップ1030において推測入力結合器137は、処理すべき単位テキストがまだあるか否かを決定する。処理すべき単位テキストがあれば、ルーチンは「YES」の分岐をたどってステップ1020に戻り、次の単位テキストを処理する。処理すべき単位テキストが無ければ、「NO」の分岐をたどってステップ1035へ進む。

【0120】ステップ1035において推測入力結合器137は、各修正候補とそれに対応する単位テキストとをワードプロセッサ210へ送る。ワードプロセッサ210はこれら候補を適切な方法で表示する。修正範囲が調整されていなければ、各候補は図6で説明したように表示される。修正範囲が单一の単位テキストまで拡張されていれば、ワードプロセッサ137は、拡張部分の追加語を、選択テキストの強調表示色とは別の強調表示色で表示しても良い。ワードプロセッサは、その単位テキストに対する候補を、図6を参照して説明したグラフィカルユーザインターフェースによって提示できる。

【0121】修正範囲モデルが複数の単位テキストを修正用に指定した場合、ワードプロセッサ210は、候補メニューをユーザーに提示し、各候補に対応する単位テキストを示すことができる。

【0122】ワードプロセッサ210がグラフィカルユーザインターフェースを介してユーザーに修正候補を提示すると、ルーチンはステップ1040において終了する。ワードプロセッサ210は、候補に対するユーザーからの応答を、図6を参照して説明したように処理する。

【0123】結論

当業者には明らかなように、本発明の精神および範囲を逸脱することなく、他の実施例も可能である。従って本発明の範囲は、前記説明に依ることなく請求の範囲によって定められる。

【図面の簡単な説明】

【図1】本発明の実施例に適した動作環境を示すブロック図である。

【図2】複数入力源データ処理システムのプログラムミュールの概略を示すブロック図である。

【図3】本発明の代表的実施例の動作を示すブロック図である。

【図4】ユーザーによる選択テキスト編集が可能な本発明の実施例を示すブロック図である。

【図5】自然言語モデルを利用可能な本発明の実施例を示すブロック図である。

【図6】複数入力源からの元データを処理するルーチンの各ステップを示すフローチャートである。

【図7】複数入力源を有する選択テキスト用候補を決定するルーチンの各ステップを示すフローチャートである。

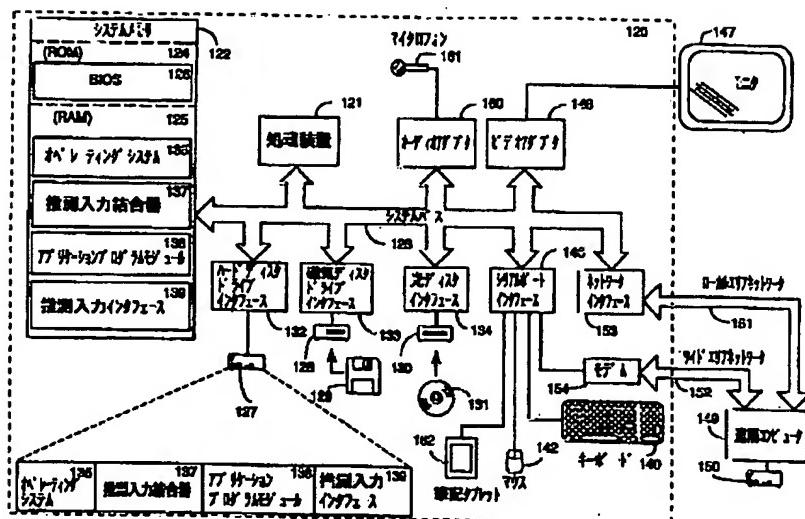
【図8】選択テキスト中のテキスト要素に対応する確率モデルを検索するためのルーチンの各ステップを示すフローチャートである。

【図9】連続確率モデルを取得するためのルーチンの各ステップを示すフローチャートである。

【図10】選択テキストの修正範囲を変更する処理を含んだオプションの元データ処理ルーチンにおける各ステップを示すフローチャートである。

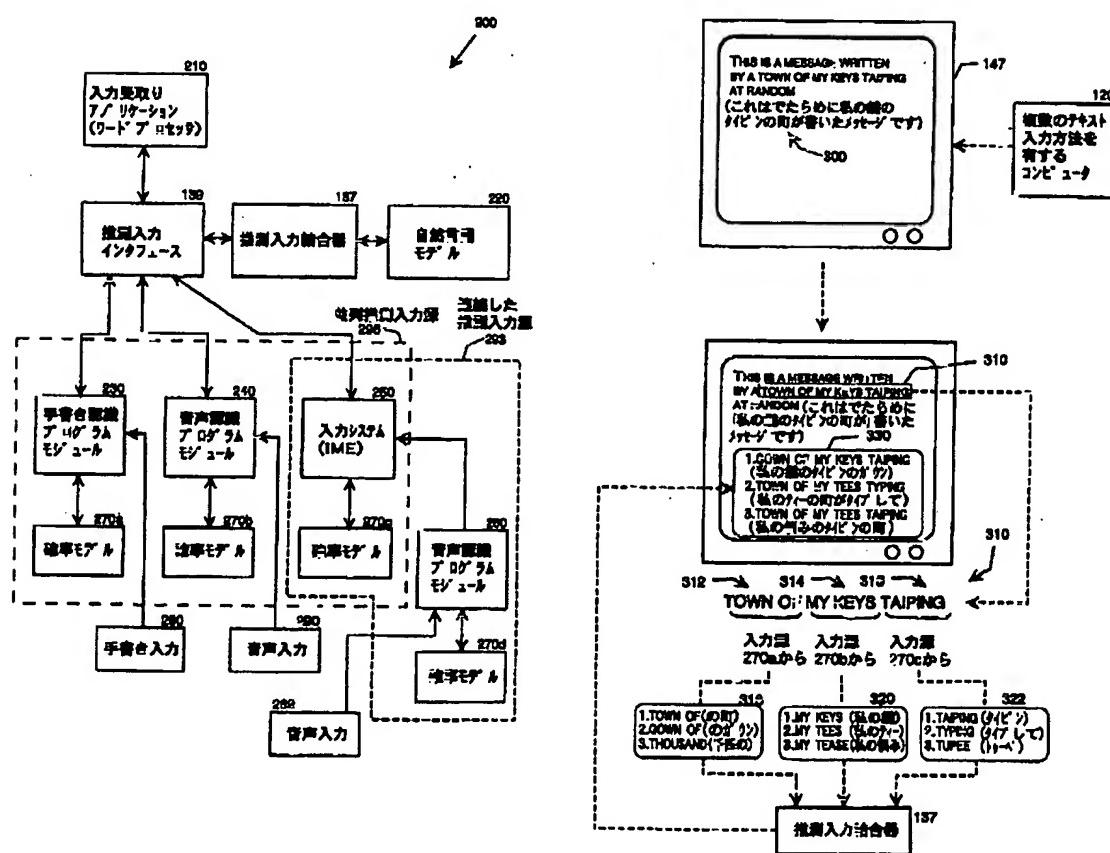
【図11】ユーザーによる選択テキスト編集を可能にする本発明の実施例動作を示すブロック図である。

【図1】

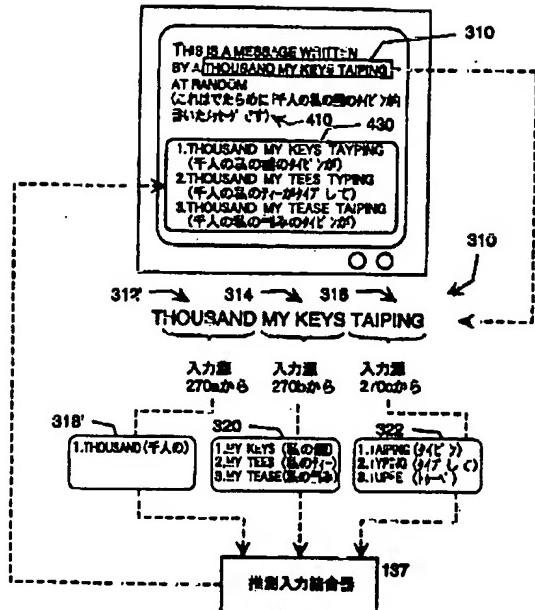


〔图2〕

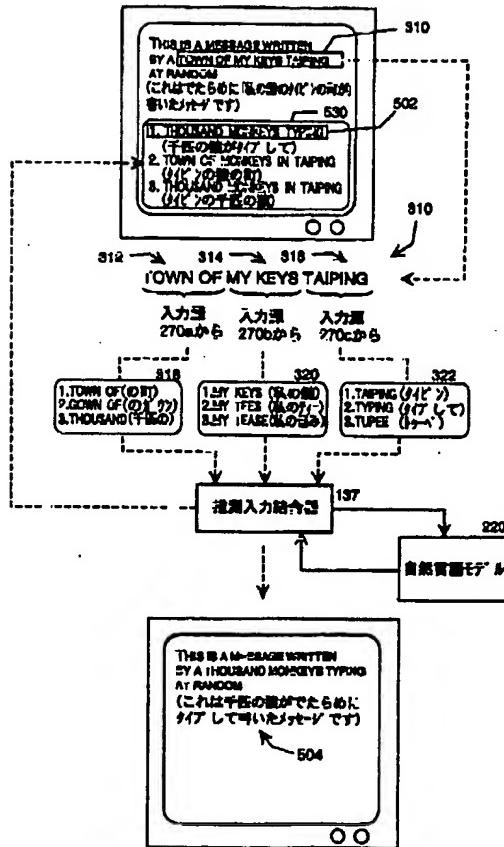
〔四三〕



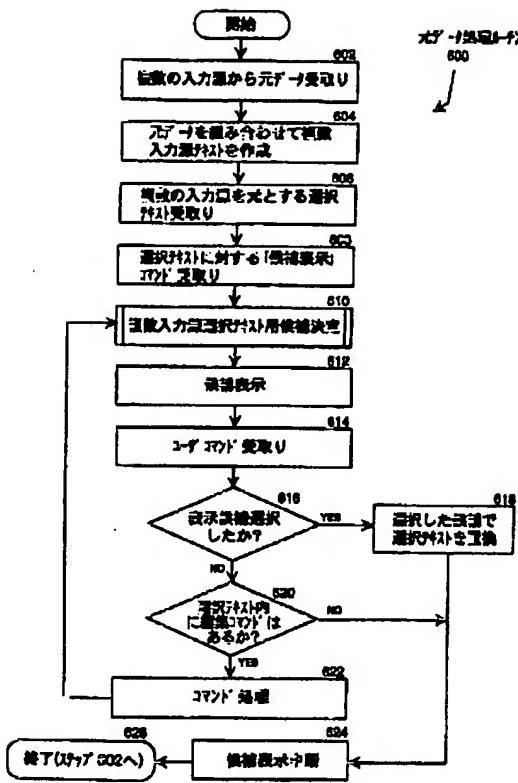
【図4】



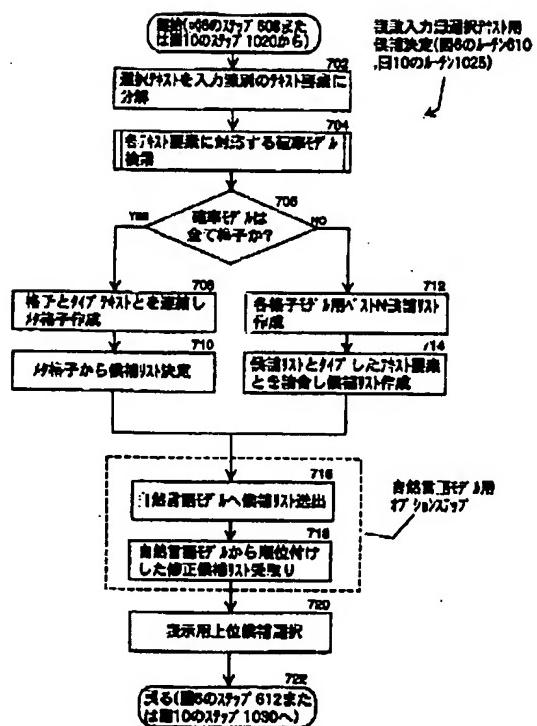
【図5】



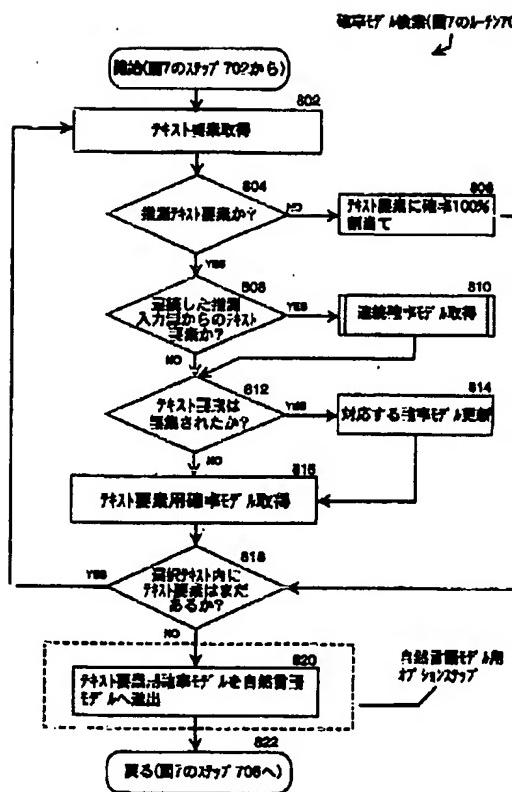
【図6】



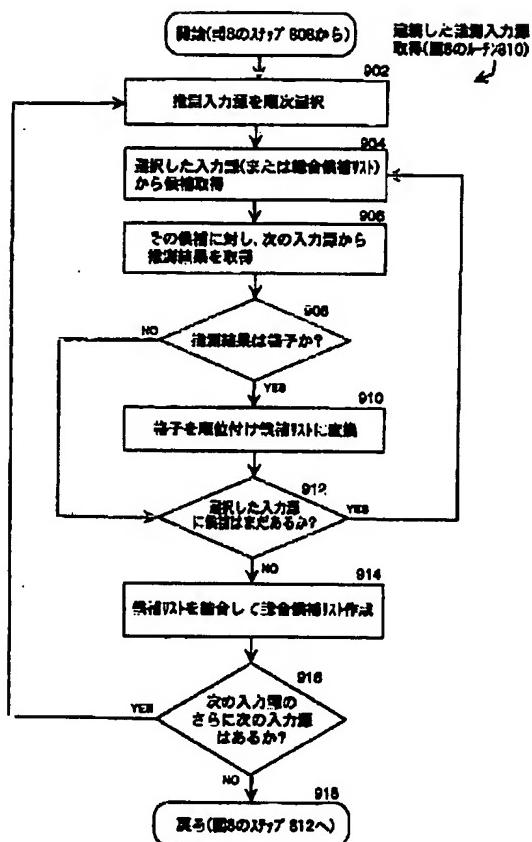
【図7】



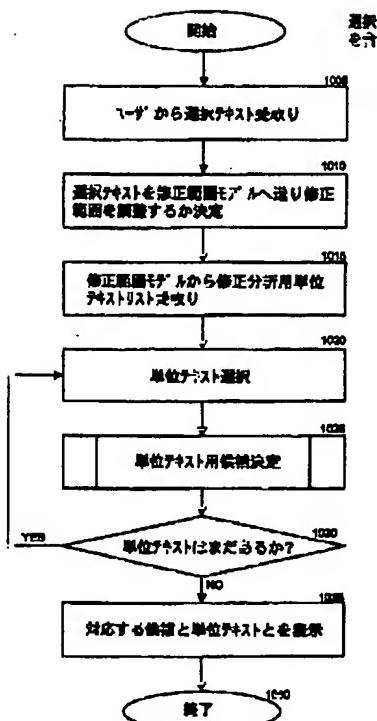
【図8】



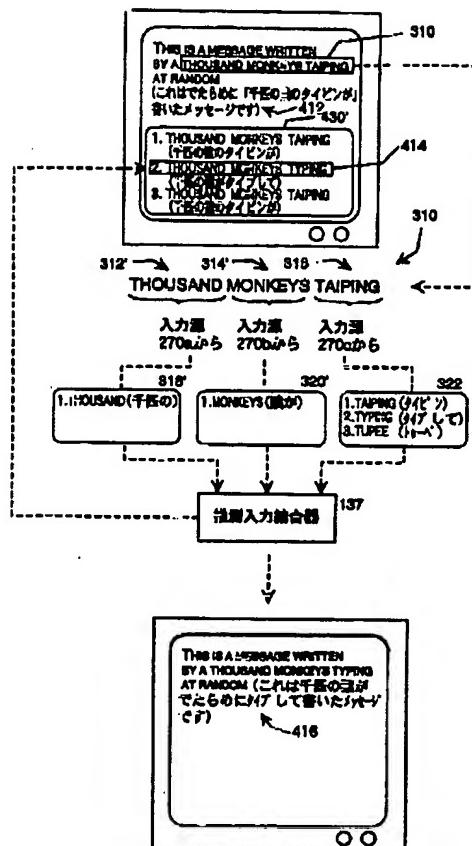
【図9】



【図10】



【図11】



フロントページの続き

(51) Int.Cl.7 識別記号
 G 10 L 15/00
 15/28
 15/22

F I
 G 10 L 3/00 (参考)
 551 P
 571 T

(72) 発明者 エリック ラッカー
 アメリカ合衆国 98105 ワシントン州
 シアトル フォーティエイス アヴェニュー
 一 4316

(72) 発明者 ポール キョン ホワン キム
 アメリカ合衆国 98107 ワシントン州
 シアトル ナンバー205 エヌダブリュー
 サーティナインス ストリート 201

【外國語明細書】

1 Title of Invention
METHOD AND SYSTEM FOR PROVIDING ALTERNATIVES FOR TEXT DERIVED FROM
STOCHASTIC INPUT SOURCES

2 Claims

(1) A computer-implemented method for correcting text, comprising the steps of:

receiving a text selection comprising a plurality of text components derived from different input sources;

at least one of the text components comprising a stochastic text component derived from a stochastic input source or a series of stochastic input sources;

receiving a command to display alternatives for the text selection;

parsing the text selection into the text components;

retrieving the stochastic model for the stochastic text component from its associated stochastic input source or series of input sources;

combining the stochastic model with other text components to produce a list of alternatives for the text selection; and

displaying the list of alternatives for the text selection on a display device.

(2) The method of claim 1, further comprising the steps of:

receiving a user command selecting one of the displayed alternatives; and

replacing the text selection with the selected alternative.

- (3) The method of claim 1, further comprising the steps of:
 - receiving an edit to the text selection;
 - producing a revised list of alternatives for the edited text selection; and
 - displaying the revised list of alternatives for the edited text selection.
- (4) The method of claim 1, further comprising the steps of:
 - receiving an edit to one of the stochastic text components;
 - retrieving a revised stochastic model for the edited stochastic text component from its associated stochastic input source or series of input sources;
 - combining the revised stochastic model with another stochastic model associated with the text selection to produce a revised list of alternatives for the edited text selection; and
 - displaying the revised list of alternatives for the edited text selection.
- (5) The method of claim 1, wherein the text selection comprises a portion of text in a file within an application selected from the group consisting of a word processor, a spreadsheet, a browser, an electronic mail program, a music transcription program, a CAD program, a presentation program, and an operating system.
- (6) The method of claim 1, wherein the step of displaying the alternatives for the text selection further comprises the steps of:
 - ranking the alternatives for the text selection in probability order; and

displaying the alternatives in their rank order on the display device.

(7) The method of claim 6, wherein the step of displaying the alternatives in their rank order further comprises the steps of:

selecting a pre-determined number of highest ranked alternatives; and

displaying the selected alternatives in their rank order on the display device.

(8) The method of claim 1, wherein the text selection comprises a plurality of stochastic text components and the step of combining the stochastic models further comprises the steps of:

combining the stochastic models for each stochastic text component to produce an interim list of alternatives for the text selection;

providing the interim list of alternatives to a natural language model;

receiving a revised list of alternatives for the text selection from the natural language model, the revised list of alternatives comprising a reevaluation of the interim list of alternatives based on natural language principles applied by the natural language model to the text selection as a whole; and

displaying the revised list of alternatives as the list of alternatives for the text selection.

(9) The method of claim 8, wherein the revised list of alternatives also comprises additional alternatives formed by the

natural language model that are not found in the interim list of alternatives provided to the natural language model.

(10) The method of claim 8, further comprising the step of providing the stochastic model retrieved for one or more stochastic text components to a natural language model for reevaluation based on natural language principles.

(11) The method of claim 8, further comprising the step of providing the stochastic model for each stochastic text component to the natural language model for use in creating the revised list of alternatives.

(12) The method of claim 1, wherein the text selection comprises a plurality of stochastic text components and the stochastic models for the text components comprise lattices, and wherein the step of combining the stochastic models to produce a list of alternatives for the text selection further comprises the steps of:

concatenating the lattices into a metalattice that includes information about any text components that are derived from a non-stochastic source; and

producing the list of alternatives for the text selection from the metalattice.

(13) The method of claim 1, wherein the text selection comprises a plurality of stochastic text components and one of the stochastic models comprises an "n-best" candidate list and another stochastic model comprises a lattice, and wherein the step of combining the stochastic models to produce a list of alternatives for the text selection further comprises the steps of:

creating an "n-best" candidate list corresponding to the lattice; and

producing the list of alternatives for the text selection by combining the "n-best" candidate lists for the text components.

(14) The method of claim 1, wherein the step of retrieving a stochastic model for a text component originating from a stochastic input source further comprises the steps:

determining if the text component is derived from stochastic input sources configured in series;

if the text component is derived from stochastic input sources configured in series, deriving a series stochastic model by combining together a stochastic model from each stochastic input source in the series;

retrieving the series stochastic model as the stochastic model for the text component.

(15) The method of claim 1, wherein the step of retrieving the stochastic model for the stochastic text component from its associated stochastic input source or series of input sources comprises the steps of:

receiving user input into a first stochastic input source in a series of stochastic input sources;

selecting a stochastic result comprising a plurality of alternatives produced by the first stochastic input source;

producing a plurality of stochastic results for a second stochastic input source in the series by using each alternative of the stochastic result produced by the first stochastic input source

as input into the second stochastic input source to produce a stochastic result for the second stochastic input source;

if any stochastic result for the second stochastic input source does not comprise an "n-best" alternatives list, converting that stochastic result to an "n-best" alternatives list;

combining the stochastic results for the second stochastic input source to create a totalized alternatives list for the second stochastic input source.

(16) The method of claim 15, wherein the step of combining the plurality of stochastic results for the second stochastic input source to create the totalized alternatives list for the second stochastic input source further comprises the steps of:

creating a single entry in the totalized alternatives list for each unique alternative appearing in the plurality of stochastic results for the second stochastic input source;

calculating a probability for each alternative in the totalized alternatives list by summing all probabilities assigned to that alternative in the plurality of stochastic results for the second stochastic input source; and

assigning each calculated probability to its associated alternative.

(17) A computer-readable medium having computer-executable instructions for performing the method of claim 1.

(18) A computer adapted to perform the method of claim 1.

(19) A computer-readable medium having computer-executable instructions for performing the method of claim 16.

- (20) A computer adapted to perform the method of claim 16.
- (21) A computer-implemented method for correcting text, comprising the steps of:
 - receiving a text selection from a user;
 - receiving a command to display alternatives for the text selection;
 - submitting the text selection to a correction scope model to determine if a scope of correction should be adjusted;
 - receiving from the correction scope model a text unit that includes the text selection and at least one adjacent word;
 - producing a list of alternatives for the text unit; and
 - displaying the list of alternatives for the text unit on a display device.
- (22) The method of claim 21, further comprising the steps of:
 - receiving a user command selecting one of the displayed alternatives; and
 - replacing the text unit with the selected alternative.
- (23) The method of claim 21, wherein the adjacent word is incorrect because of a related error that caused a word within the text selection to be incorrect.
- (24) The method of claim 21, wherein the correction scope model includes criteria selected from the group consisting of a natural language model, a model of likely errors, an acoustic model, a handwriting model, and a vision-based model.

(25) The method of claim 21, wherein the step of producing a list of alternatives for the text unit further comprises the steps of:

parsing the text unit into text components derived from different input sources;

determining if one of the text components comprises a stochastic text component;

if the one of the text components comprises the stochastic text component, retrieving a stochastic model for the stochastic text component; and

combining the stochastic model with other text components to produce a list of alternatives for the text unit.

(26) A computer-readable medium having computer-executable instructions for performing the method of claim 21.

(27) A computer adapted to perform the method of claim 25.

3 Detailed Description of Invention

FIELD OF THE INVENTION

The invention relates generally to methods for entering text into a computer and, more particularly, relates to providing alternatives for a text selection derived from multiple stochastic input sources.

BACKGROUND OF THE INVENTION

Computer users have traditionally entered text into word processors through a keyboard and mouse. In recent years, however, word processors have become more sophisticated by allowing users to enter text into them through other input methods, such as speech or handwriting. Although a computer cannot always interpret such input

with complete accuracy, a computer can generate a list of text alternatives for the input. Furthermore, the computer can often assign to each alternative a probability that the alternative is the one the user intended. Input that produces such probabilistic results is called "stochastic input," while input that can be accurately determined, such as typed text, is called "non-stochastic input."

Typically, text produced in a word processor from stochastic input must be heavily edited by the user in order to produce the text intended by the user when he or she created the stochastic input. The editing process has been simplified by allowing the user to select text created from stochastic data and request alternatives for the text selection. In response, the computer can provide the user with alternatives for the text selection through a graphical user interface. If the user chooses one of the alternatives, the computer replaces the text selection with the selected alternative.

Suppose that after a user creates text in a word processor by providing the word processor with stochastic input, such as speech, the user then edits the text. The user may, for example, replace a word of the text with a new word typed into the computer with a keyboard. Current word processors do not incorporate typed text edits into the alternatives they provide for an edited text selection. Thus, there is a need in the art for a method of providing alternatives to edited text derived from stochastic input.

Another problem occurs if the user attempts to request alternatives for a text selection spanning multiple stochastic input sources. For instance, the user may request alternatives for a text selection containing a word based on handwriting input and a word based on speech input. Current word processors are not capable of

providing meaningful alternatives for such a text selection. Thus, there is also a need in the art for a method of providing alternatives for a text selection derived from multiple stochastic input sources.

An input method editor (IME) is another word processor input method that produces stochastic data. Generally, an IME converts input into foreign language text. The input into an IME may, for example, be typed text entered into the computer through a keyboard and mouse. An IME is especially useful for creating ideograms in Asian and other languages. Because there are many more ideograms in such languages than there are keys on a keyboard, entering a particular ideogram into the computer typically requires multiples keystrokes, which the IME interprets as a composed character.

In a typical IME, a user may type in English characters defining a phonetic spelling for a desired Chinese character. Since many Chinese characters have similar pronunciations, the typed phonetic spelling may represent any one of a number of different Chinese characters. The IME then provides the user with the most probable candidates corresponding to the typed phonetic spelling so that the user can choose the correct one.

Programmers have previously recognized the value of providing speech input into an IME. This is done by first converting the speech into text, which is then used as input into the IME. As has already been explained, however, the interpretation of speech is stochastic in nature. Hence, the text produced by the speech interpreter may not be the text that was intended by the user. If incorrect text is used as input into the IME, the results produced by the IME are likely to be poor. Accordingly, when speech is used as input into an IME, the program interpreting the speech data

typically allows the user to first correct the text produced by the speech interpreter before inputting that text into the IME. When the IME produces foreign language translations of the text, the user may again choose the desired alternative because the result of an IME is also stochastic in nature. Requiring the user to edit the results at two different stages of the process can be inefficient and inconvenient. Thus, there is a further need in the art for an improved method of handling speech input to an IME.

SUMMARY OF THE INVENTION

The present invention meets the needs described above in a stochastic input combiner that facilitates the editing of text. The invention does this by providing alternatives for a text selection made by the user, even where that text selection is derived from multiple input sources, one or more of which can be stochastic in nature.

The stochastic input combiner provides the alternatives to the user in list form through a graphical user interface. The user can then choose one of the alternatives to replace the text selection the user has highlighted for editing. This can often be quicker than requiring the user to think of alternatives on his or her own and then make changes accordingly using a keyboard and mouse. If the user does not find an alternative the user likes for the text selection, the user can edit the text selection using the keyboard and mouse. In response, the stochastic input combiner can provide an updated list of alternatives that incorporate the user's changes. Often, the user need only partially edit the text selection before the stochastic input combiner produces an alternative the user likes, so the stochastic input combiner again improves editing efficiency.

The stochastic input combiner may also provide the advantage of a natural language model. Such a model may analyze the text selection as a whole using natural language principles in order to provide a better list of alternatives for the text selection. This, too, improves the user's editing efficiency.

Generally described, the present invention includes a method for correcting text. The user first enters text into the computer, perhaps using multiple stochastic input sources. The user may also use a keyboard and mouse to enter text into the computer.

Keyboard/mouse entry of text is an example of an input source which is non-stochastic, meaning that the text intended to be entered through the source can be determined with complete accuracy. On the other hand, a stochastic input source is one that converts input into a list of alternatives, each having less than 100% probability of being the correct alternative. Because speech cannot always be interpreted by a computer with complete accuracy, a speech recognition unit is an example of a stochastic input source which converts speech input into a list of text alternatives. Other examples of a stochastic input source include a handwriting recognition unit and an input method editor (IME). Where an input source for a text component is stochastic, the most likely alternative for the text component is generally used to represent the text component in the text selection.

Once the user enters text into the computer, the user can begin the correction process by making a text selection of a portion of the text the user entered. This text selection can include multiple text components. Such a text component can be a subset of the text selection that the user entered through a single input

source. The user may have entered different text components within the text selection using different input sources, and the user may have entered one or more of the text components with a stochastic input source.

Once the user makes a text selection, the user may enter a command to display alternatives for the text selection as a whole. The stochastic input combiner then parses the text selection into its text components and retrieves a stochastic model representing the alternatives for a text component originating from a stochastic input source. This stochastic model may include a list of the alternatives for the text component together with the probabilities associated with the alternatives. Alternatively, the stochastic model may include a lattice.

The stochastic input combiner then combines the stochastic model retrieved with other text components to produce a list of alternatives for the text selection as a whole. The stochastic input combiner then displays this list of alternatives for the text selection on a display device, such as a monitor. The user may then select one of the displayed alternatives. In that case, the selected alternative replaces the text selection.

The stochastic input combiner may also utilize a natural language model. In this alternative, the stochastic input combiner may combine the stochastic models for each stochastic text component to produce an interim list of alternatives for the text selection that the combiner provides to the natural language model. The natural language model forms a revised list of alternatives by reevaluating the interim list of alternatives based on natural language principles applied by the natural language model to the text selection as a whole. The natural language model may also add

new alternatives to the revised list of alternatives that are not found in the interim list. After the natural language model returns the revised list of alternatives to the stochastic input combiner, the stochastic input combiner provides the revised list of alternatives for display.

In another alternative, a series of stochastic input sources creates a stochastic text component. This means that at least one stochastic input source produces a stochastic result that serves as input into a second stochastic input source. Typically, the first stochastic input source of the series requires user input, while subsequent stochastic input sources in the series receive an alternative produced by the previous stochastic input source as input. An example of this is a speech recognition unit that produces text used as input into an IME. When stochastic input sources in series produce a stochastic text component, the stochastic input combiner can produce a series stochastic model that contains alternatives for the stochastic text component and accurate probabilities for those alternatives that account for the likelihood of the inputs into each stochastic input source in the series. This process eliminates any need for the user to choose a single alternative produced by a stochastic input source to use as input into a subsequent stochastic input source.

To produce a series stochastic model for a text component derived from a series of stochastic input sources, the stochastic input combiner first submits to the first stochastic input source in the series the user input intended for that stochastic input source. By processing the user input, the first stochastic input source produces a stochastic result. The stochastic result that the first stochastic input source produces has multiple alternatives, and the

stochastic input combiner selects that stochastic result. By using each alternative of the selected stochastic result as input into the second stochastic input source to produce a stochastic result for the second stochastic input source, the stochastic input combiner produces multiple stochastic results, each stochastic result having multiple alternatives, for the second stochastic input source. If any stochastic result for the second stochastic input source does not contain an "n-best" alternatives list, the stochastic input combiner converts that stochastic result into an "n-best" alternatives list because converting all stochastic results into the same format simplifies the process of combining them. The stochastic input combiner then combines the stochastic results for the second stochastic input source to create a totalized alternatives list for the second stochastic input source. If there are only two stochastic input sources in the series, then the totalized alternatives list may serve as the stochastic model for the text component resulting from the series.

The stochastic input combiner may also be functional for expanding the scope of correction for a text selection received from a user to a larger text unit. To do this, the stochastic input combiner submits the text selection to a correction scope model to make the determination of whether the scope of correction should be adjusted. In response to submitting the text selection, the stochastic input combiner receives from the correction scope model a text unit that includes the text selection and at least one adjacent word. Using the text unit, the stochastic input combiner can then produce a list of alternatives for the text unit and display those alternatives on a display device.

The various aspects of the present invention may be more clearly understood and appreciated from a review of the following detailed description of the disclosed embodiments and by reference to the appended drawings and claims.

DETAILED DESCRIPTION OF THE EXEMPLARY EMBODIMENTS

The present invention is typically embodied in a word processor that can receive input from multiple sources, each of which can be a non-stochastic input source or a stochastic input source. Keyboard/mouse entry of text is an example of an input source which is non-stochastic, meaning that the computer can determine the text the user intended with complete accuracy. On the other hand, a stochastic input source is one that converts input into a stochastic result. A stochastic result is one having multiple alternatives, each having less than 100% probability of being the correct alternative. An example of a stochastic input source is a speech recognition unit, which converts speech input into a list of text alternatives since a computer cannot always interpret speech with complete accuracy. Other examples of a stochastic input source are a handwriting recognition unit and an input method editor (IME).

The word processor is functional for allowing the user to select a section of text and to request alternatives for that selection. If the computer has created the text selection from one or more stochastic input sources, there will be alternatives for the text selection. After the word processor determines alternatives for the text selection, the word processor displays the alternatives through a graphical user interface. If the user chooses one of the

alternatives for the text selection, then the word processor replaces the text selection with the chosen candidate.

After examining the list of alternatives that the word processor provides, the user may not find an acceptable alternative for the text selection. Hence, the word processor may allow the user to edit the text selection using a keyboard and mouse. The user may, for example, change one of the words in the text selection. In that case, the word processor may then revise the list of alternatives to incorporate the edit and provide the revised list of alternatives to the user. If the user chooses one of the revised alternatives for the text selection, then the word processor replaces the text selection with the chosen alternative.

A program module called a stochastic input combiner is typically responsible for producing the alternatives for a text selection. The stochastic input combiner does this by parsing the text selection into smaller text components derived from no more than one stochastic input source. For each stochastic text component, the stochastic input combiner then retrieves a stochastic model representing the alternatives for the text component. Then, the stochastic input combiner can combine the stochastic models retrieved with other text components to produce a list of alternatives for the text selection as a whole.

The stochastic input combiner can be part of the word processing application. Alternatively, the stochastic input combiner can be a separate utility that is part of the operating system. The combiner could also be a separate program that interfaces with the word processor but that is not part of the operating system.

To improve the list of alternatives the word processor offers for a text selection, the word processor may use a natural language model. The natural language model may apply natural language principles to the text selection as a whole to reevaluate the likelihood of the alternatives in the alternatives list produced by the stochastic input combiner and to add new alternatives to the alternatives list. Alternatively, the natural language model may apply natural language principles to individual text components to improve the stochastic models that the stochastic input combiner uses to produce the alternatives list.

A text component is occasionally derived from a series of stochastic input sources. This means that at least one stochastic input source produces a stochastic result that serves as input into a second stochastic input source. The first stochastic input source in the series typically receives user input, while the last stochastic input source in the series produces the alternatives for the text component. The stochastic input combiner can derive a series stochastic model containing alternatives for such a text component without requiring the user to select a single alternative result for each stochastic input source in the series as input for the subsequent stochastic input source.

To derive a series stochastic model for a text component, the stochastic input combiner first selects the user input that ultimately produced the text component. The stochastic input combiner then submits the selected user input to the first stochastic input source in series order. The stochastic input combiner then submits each alternative produced by the first stochastic input source as input into the subsequent stochastic input source in series order. Because the subsequent stochastic

input source produces a stochastic result from each alternative submitted to it, the stochastic results of the subsequent stochastic input source must be combined into a totalized candidate list. If there is yet another stochastic input source in the series, the stochastic input combiner submits each candidate of the totalized candidate list as input into that next stochastic input source. The totalized candidate list that the last stochastic input source of the series produces in this fashion is the series stochastic model.

Often, stochastic input sources make an error which spans multiple words. During the correction process, a user may not notice the full extent of that error. For example, if the user dictates the word "recognize," the speech recognition engine might conclude that the most likely interpretation of the speech input is "wreck a nice." While editing, the user might see the word "wreck" and request alternatives only for that word because the user did not notice that the following words were also incorrect (i.e., "a nice").

If a user makes a text selection that does not include adjacent words that are incorrect because of a related error and if the word processor only uses the user's text selection to produce an alternatives list, none of the alternatives offered for the text selection may be the text the user intended at the time of input. Similarly, replacing the text selection with an alternative chosen by the user from the alternatives list would leave the incorrect adjacent words in the text.

To eliminate these disadvantages, the stochastic input combiner may submit the text selection to a correction scope model which determines if the scope of correction should be expanded. In the "recognize" example, an appropriate text unit for correction

would be "wrcck a nice." To make this determination, the correction scope model may draw on information included in a natural language model, models of likely errors, and models tied to the input methods used to produce the text in the word processor. The models associated with the input methods may include acoustic models for speech recognition, handwriting models for handwritten input, and vision-based models for recognizing sign language or other gestures.

If the correction scope model determines that the scope of correction should be adjusted, the correction scope model identifies one or more larger text units for which the stochastic input combiner should produce alternatives in the manner already described. The correction scope model sends a list of these text units to the stochastic input combiner for processing.

Turning now to the figures, in which like numerals refer to like elements throughout the several figures, aspects of the present invention will be described.

Exemplary Operating Environment

FIG. 1 and the following discussion are intended to provide a brief and general description of a suitable computing environment 100 for an implementation of the present invention. The exemplary operating environment 100 includes a conventional personal computer system 120, including a processing unit 121, a system memory 122, and a system bus 123 that couples the system memory 122 to the processing unit 121. The system memory 122 includes read only memory (ROM) 124 and random access memory (RAM) 125. A basic input/output system 126 (BIOS), containing the basic routines that

help to transfer information between elements within the personal computer system 120, such as during start-up, is stored in ROM 124.

The personal computer system 120 further includes a hard disk drive 127, a magnetic disk drive 128, e.g., to read from or write to a removable magnetic disk 129, and an optical disk drive 130, e.g., for reading a CD-ROM disk 131 or to read from or write to other optical media. The hard disk drive 127, magnetic disk drive 128, and optical disk drive 130 are connected to the system bus 123 by a hard disk drive interface 132, a magnetic disk drive interface 133, and an optical drive interface 134, respectively. The drives and their associated computer-readable media provide nonvolatile storage for the personal computer system 120. Although the description of computer-readable media above refers to a hard disk, a removable magnetic disk and a CD-ROM disk, it should be appreciated by those skilled in the art that other types of media that are readable by a computer system, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, and the like, may also be used in the exemplary operating environment.

A user may enter commands and information into the personal computer 120 through conventional input devices, including a keyboard 140 and pointing device, such as a mouse 142. A microphone 161 may be used to enter audio input, such as speech, into the computer system 120. A user may enter graphical information, such as drawings or handwriting, into the computer system by drawing the graphical information on a writing tablet 162 using a stylus. The computer system 120 may include additional input devices (not shown), such as a joystick, game pad, satellite dish, scanner, touch screen/stylus, or the like. The microphone 161 can be connected to the processing unit 121 through an audio adapter 160 that is coupled

to the system bus. The other input devices are often connected to the processing unit 121 through a serial port interface 146 that is coupled to the system bus, but may be connected by other interfaces, such as a game port or a universal serial bus (USB).

A monitor 147 or other type of display device is also connected to the system bus 123 via an interface, such as a video adapter 148. In addition to the monitor, personal computer systems typically include other peripheral output devices (not shown), such as speakers or printers.

The personal computer system 120 may operate in a networked environment using logical connections to one or more remote computer systems, such as a remote computer system 149. The remote computer system 149 may be a server, a router, a peer device or other common network node, and typically includes many or all of the elements described relative to the personal computer system 120, although only a memory storage device 150 has been illustrated in FIG. 1. The logical connections depicted in FIG. 1 include a local area network (LAN) 151 and a wide area network (WAN) 152. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the personal computer system 120 is connected to the LAN 151 through a network interface 153. When used in a WAN networking environment, the personal computer system 120 typically includes a modem 154 or other means for establishing communications over a WAN 152, such as the Internet. The modem 154, which may be internal or external, is connected to the system bus 123 via the serial port interface 146. In a networked environment, program modules depicted relative to the personal computer system 120, or portions thereof, may be stored in

the remote memory storage device 150. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computer systems may be used. It will be further appreciated that the invention could equivalently be implemented on host or server computer systems other than personal computer systems, and could equivalently be transmitted to the host computer system by means other than a CD-ROM, for example, by way of the network connection interface 163.

A number of program modules may be stored in the drives and RAM 125 of the computer system 120. Program modules control how the computer system 120 functions and interacts with the user, with I/O devices or with other computers. Program modules include routines, operating system 135, application program modules 138, data structures, browsers, and other software or firmware components. The invention may conveniently be implemented in one or more program modules, such as a stochastic input combiner program module 137 and a stochastic input interface program module 139, each of which is based upon the methods described in the detailed description.

The application program modules 138 may comprise a variety of applications used in conjunction with the present invention, some of which are shown in FIG. 2. The purposes of and interactions between some of these program modules are discussed more fully in the text describing FIG. 2. These include a word processor program 210 (such as WORD, produced by Microsoft Corporation of Redmond, WA), a handwriting recognition program module 230, a speech recognition program module 240, and an input method editor (IME) 250.

No particular programming language will be described for carrying out the various procedures described in the detailed description because it is considered that the operations, steps, and

procedures described and illustrated in the accompanying drawings are sufficiently disclosed to permit one of ordinary skill in the art to practice an exemplary embodiment of the present invention. Moreover, there are many computers and operating systems which may be used in practicing an exemplary embodiment, and therefore no detailed computer program could be provided which would be applicable to all of these many different systems. Each user of a particular computer will be aware of the language and tools which are most useful for that user's needs and purposes.

Those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

Overview of Program Modules

FIG. 2 provides an overview of the program modules of a multi-source data processing system 200. Generally, the program modules shown in FIG. 2 enable a user to enter text into an application 210, such as a word processor, using both stochastic and non-stochastic input sources. Typical stochastic input sources include a handwriting recognition program module 230, speech recognition program module 240, input method editor (IME) 250, and speech recognition program module 260. A keyboard 140 is a typical source of non-stochastic data. Once the user enters text into the

word processor 210 through one or more of these input sources, the user may then select a section of text and request a candidate list of alternatives for that text selection. The text selection may contain input from multiple stochastic and non-stochastic input sources. As long as the text selection is derived from at least one stochastic input source, there will be alternatives for the text selection. The program modules can produce this candidate list and present them to the user through a graphical user interface. If the user chooses one of the candidates, the text selection is replaced with the chosen candidate. The operation of stochastic input sources 230, 240, 250, and 260 are now discussed in turn.

The handwriting recognition program module 230 receives handwriting input 280 from the user. The user may generate the handwriting input 280 by writing on the writing tablet 162 with a stylus. Alternatively, the user may generate handwriting input 280 using other devices. For instance, the user may write on the monitor 147 with a mouse 142, or the user may write on a touch screen using a stylus. After input, the handwriting input 280 is preferably directed to the handwriting recognition program module 230 by a writing tablet driver module in the operating system 136.

As handwriting is often difficult for a computer to interpret, the handwriting recognition program module 230 cannot always decipher the handwriting input 280 with complete accuracy. The best the program module 230 can do is to generate alternatives for the handwriting input 280 and assign a probability that each alternative is the correct one. By definition, then, the handwriting recognition program module 230 generates a stochastic result. The stochastic model 270a includes a data structure containing the

stochastic data produced by processing handwriting input 280 with the handwriting recognition program module 230.

Although any data structure capable of storing stochastic data can comprise a stochastic model 270, two useful structures for doing so are a lattice and an "n-best" alternatives list. A lattice is a structure that is well known to those skilled in the art, so a complete description will not be given. Briefly, however, a lattice stores words or phrases produced by a stochastic input source in nodes. Because each word or phrase is stochastic data, the node also stores a probability assigned to the associated word or phrase. Using methods well known to those skilled in the art, the lattice can be traversed in order to produce likely alternatives for any section of text represented by the stochastic data. Furthermore, lattices representing adjacent pieces of text can be combined into a metalattice through a process known as concatenation. The metalattice can then be traversed to produce alternatives for the adjacent pieces of text.

Alternatively, stochastic data may be represented by a list of the n-best alternatives and their associated probabilities. For any given word or phrase, an n-best alternatives list may be produced from a lattice representing the word or phrase.

The speech recognition program module 240 works like the handwriting recognition program module 230, except that it receives speech input 290 from the user through a microphone 161 run by a microphone driver module in the operating system 13b. Speech is often difficult to interpret because many words that sound alike have different meanings and spellings, so the speech recognition program module 240 also produces a stochastic result. The stochastic model 270b stores the data structure containing the

stochastic data produced by processing speech input 290 with the speech recognition program module 240.

An input method editor (IME) 250 also generates stochastic data. Generally, an IME 250 converts input into foreign language text. The input into an IME 250 may, for example, be typed text entered into the computer through a keyboard 140 and mouse 142. The stochastic model 270c includes a data structure containing the stochastic data produced by the IME 250.

An IME 250 is especially useful for creating ideograms in Asian and other languages. Because there are many more ideograms in such languages than there are keys on a keyboard, entering a particular ideogram into the computer is problematic without an IME 250. In a typical IME 250, a user types in English characters a phonetic spelling for a desired Chinese character. Since many Chinese characters have similar pronunciations, the typed phonetic spellings may represent any one of a number of different Chinese characters, and the IME 250 produces a stochastic result. The IME 250 then provides the user with the most probable candidates intended by the typed phonetic spelling so that the user can choose the correct one.

The stochastic results produced by one stochastic input source may serve as stochastic input to a second stochastic input source. When this is the case, the stochastic input sources are "series stochastic input sources," and the stochastic input sources can be described as configured "in series." This is illustrated by the configuration 293 of program modules, which also demonstrates another embodiment of an IME 250.

In this embodiment, English speech input 262 may be entered into the computer and used to produce Japanese text. The speech 262

is first submitted to a speech recognition program module 260. In operation, the speech recognition program module 260 functions much like the speech recognition program module 240, but it is illustrated as a distinct unit because it may have a different speech interpretation engine. For example, the speech recognition program module 260 may interpret a different language than the speech recognition program module 240. The stochastic model 270d includes a data structure containing the stochastic data produced by processing speech input with the speech recognition program module 260.

In an English speech/Japanese IME example, the speech recognition program module 260 may produce English text alternatives from the spoken English words and store them in the stochastic model 270d. One or more of the English language text alternatives stored in the stochastic model 270d can then be used as input into the IME 250, which translates the English language text input into Japanese characters. Each alternative input into the IME 250 produces a separate stochastic result, though it should be appreciated that there may be overlap between the alternatives forming the stochastic results of two distinct inputs into the IME 250.

Though the arrow in FIG. 2 from the speech recognition program module 260 to IME 250 illustrates that the speech recognition program module is a stochastic input source for the IME 250, it should be understood that the two program modules may not interface directly. Thus, for example, stochastic input from speech recognition program module 260 to IME 250 may travel through an interface program module, such as stochastic input interface 139, to which each stochastic input source is directly connected.

A stochastic input interface 139 serves as a conduit for stochastic data between an application 210 that is to receive stochastic data and a stochastic input source, such as handwriting recognition program module 230, speech recognition program module 240, or IME 250. One advantage of having a stochastic input interface 139 as a conduit for stochastic data is that it simplifies communication between the application 210 receiving the stochastic data and the stochastic input sources. That is, the application only needs to know how to communicate with the stochastic input interface instead of all possible stochastic input sources. The application 210 that is to receive stochastic input is a word processor in an exemplary embodiment of the present invention. However, the application 210 could also be a spreadsheet, browser, electronic mail program, music transcription program, CAD program, presentation software (such as PowerPoint, produced by Microsoft Corporation of Redmond, Washington), operating system, or other software program.

In the word processor embodiment, the word processor 210 receives, through stochastic input interface 139, text representing the most likely alternative from each stochastic input source used to enter data into the word processor. In addition to transmitting data into the word processor 210 through multiple stochastic input sources, the user may also enter typical non-stochastic data into the word processor, such as by typing on a keyboard 140. The word processor 210 combines all this source data into a multi-source text string that is presented to the user. Although the word processor 210 does not indicate to the user the source of each word of the text, the word processor nonetheless maintains a record of the source of each component of the text.

The word processor 210 is also functional for allowing the user to identify a section of text and to request alternatives for that selection. If the text selection is derived from one or more stochastic input sources, there will be alternatives for the text selection. The word processor 210 can request a candidate list of alternatives from the stochastic input interface 139 by providing it with the text selection and the sources of each of the components of that text selection. After the request is processed, the stochastic input interface 139 provides the word processor 210 with a candidate list for the entire text selection. The word processor 210 provides the candidate list to the user through a graphical user interface. If the user chooses one of the alternatives for the text selection from the candidate list, then the word processor replaces the text selection with the chosen candidate.

In order to process the request for a candidate list of alternatives for a text selection, the stochastic input interface 139 transmits the request to the stochastic input combiner 137. By communicating with the stochastic input sources through the stochastic input interface 139, the stochastic input combiner 137 can retrieve information about the stochastic models 270 needed to produce the candidate list for the text selection.

To produce the candidate list, the stochastic input combiner 137 may optionally consult a natural language model 220. To do so, the combiner 137 first produces an interim candidate list of alternatives for the text selection using the information retrieved from the stochastic models 270. After the combiner 137 provides the interim candidate list to the natural language model 220, the natural language model analyzes the interim candidate list using clues such as grammar, the overall meaning of a section of text, and

the probability of various word sequences. Based upon this analysis, the natural language model 220 produces additional alternatives for the candidate list and reevaluates the probabilities of those alternatives in the interim candidate list. The methods used to produce a candidate list of alternatives for a text selection will be described with reference to FIGS. 3-9 and 11.

As shown in FIG. 2, stochastic input sources 230, 240, and 250 can each provide stochastic data to word processor 210 without first filtering their stochastic data through another stochastic input source. In other words, stochastic input sources 230, 240, and 250 can each directly (through stochastic input interface 139) transmit stochastic data to the word processor 210, and stochastic data from each source can be incorporated into the same word processing document. For this reason, they are "parallel stochastic input sources" 296, and these stochastic input sources may be described as configured "in parallel."

Although the various program modules have been described separately, one skilled in the art should recognize that the modules could be combined in various ways and that new program modules could be created to accomplish similar results. In particular, the stochastic input combiner 137 and the natural language model 220 could reside in the stochastic input interface 139, and all three program modules could be part of the operating system 135 or the word processor 210. Also, the combiner 137 and the natural language model 220 could be separate programs that interface with the word processor 210 directly. Similarly, the stochastic input sources 230, 240, 250, and 260 could be stand-alone application program modules 138, or they could be part of the operating system 135.

Graphical Illustration of a Typical Implementation

FIGS. 3 5 and 11 illustrate what the user sees and does in a typical implementation of the present invention. Furthermore, these figures show the functionality of the stochastic input combiner 137 and the natural language model 220.

In FIG. 3, a computer 120 with multiple text entry methods accepts input from a user and transmits that input to an application 210, such as a word processor. The computer converts that input into a text string 300, which it displays on monitor 147. In this example, the user intended to produce the text "THIS IS A MESSAGE WRITTEN BY A THOUSAND MONKEYS TYPING AT RANDOM." However, the computer interpreted the stochastic input as "THIS IS A MESSAGE WRITTEN BY A TOWN OF MY KEYS TAIPING AT RANDOM" to produce text 300.

Once the text 300 is displayed, the user may make a text selection 310 by highlighting a portion of the text. The text selection 310 in FIG. 3 has three text components 312, 314, and 316. Each of the text components 312, 314, and 316 comes from a different stochastic input source. Thus, for example, text component 312 may be one of the alternatives produced by processing handwriting input 280 with the handwriting recognition program module 230. The alternatives produced by this stochastic input source are stored in the stochastic model 270a. The alternatives list 318 is a list of the alternatives for text component 312 stored in the stochastic model 270a. Furthermore, the computer has chosen "TOWN OF" for text component 312 because the computer determined that that is the most likely alternative in alternatives list 318 for the handwriting input 280 that was used to produce the text component 312. As shown by alternatives list 318, however, the

computer also recognizes that "TOWN OF" and "THOUSAND" are possible alternative phrases for text component 312.

Similarly, speech recognition program module 240 produces the stochastic result stored in stochastic model 270b by processing speech input 290. The alternatives list 320 contains the alternatives stored in stochastic model 270b for text component 314.

"MY KEYS" has been selected for text component 314 because it is the most likely alternative.

Likewise, text component 316 "TAIPING" comes from yet a third stochastic input source. This stochastic input source stores its alternatives in stochastic model 270c, and the alternatives are represented in list form in alternatives list 322.

The stochastic input combiner 137 forms various combinations of alternatives from alternatives list 318, 320, and 322. The stochastic input combiner 137 then produces a ranked list of the various combinations it has produced based on its calculation of the probability that each combination is the one intended by the user for text selection 310. The top ranked alternatives for text selection 310 are then displayed in alternatives list 330 on the monitor 147.

After the alternatives list 330 for the text selection is displayed, the user may choose to edit the text selection, as shown in FIG. 4. In that figure, the user has made edit 410 by typing the word "THOUSAND" over the words "TOWN OF" from FIG. 3. As a result, text component 312 of text selection 310 is replaced with text component 312'.

Edit 410 may have been accomplished by using a keyboard 140 and a mouse 142. Because such an entry method is non-stochastic in nature, there are no alternatives for text component 312'. This

change is reflected in alternatives list 318', which has replaced alternatives list 318 from FIG. 3. The only alternative shown in alternatives list 318' is "THOUSAND."

After the edit is completed, stochastic input combiner 137 again forms various combinations of alternatives from alternatives lists 318', 320, and 322 in order to form alternatives for the edited text selection 310. These alternatives are displayed as alternatives list 430 on monitor 147.

If the user desires a different alternative to replace text selection 310 than the alternatives displayed in alternatives list 430, the user may again edit text selection 310. This is shown in FIG. 11, with the user making edit 412 by typing the word "MONKEYS" over the phrase "MY KEYS" that appeared in text selection 310 in FIG. 4. As a result, text component 314, "MONKEYS," replaces text component 314, "MY KEYS." This further results in the replacement of alternatives list 320 with alternatives list 320', which contains only one alternative because an edit is non-stochastic in nature. Once again, the stochastic input combiner 137 combines alternatives from the various alternatives lists 318', 320', and 322 in order to produce alternatives list 430' for text selection 310.

At that point, the user may find an alternative 414 which he wishes to replace text selection 310. If so, he may highlight and choose alternative 414, in which case alternative 414 replaces the text selection 310 to produce new text 416.

FIG. 5 is similar to FIG. 3, except that FIG. 5 illustrates an embodiment of the invention in which the natural language model 220 is operative. As in FIG. 3, a user makes text selection 310 comprised of text components 312, 314, and 316. The stochastic

input combiner 137 then forms various combinations of alternatives for the text components to produce an interim list of alternatives for text selection 310. Instead of displaying the most probable alternatives thus derived, the stochastic input combiner 137 then passes the interim list of alternatives for the text selection 310 to a natural language model 220.

The natural language model 220 then re-evaluates the probabilities of the various alternatives in the interim alternatives list based on natural language principles applied by the natural language model to the text selection 310 as a whole. This includes analysis of grammatical and other language clues in the text selection. The natural language model 220 may also form additional alternatives for the text selection 310 not found in the interim list of alternatives provided to the natural language model 220. The natural language model 220 returns to the stochastic input combiner 137 a revised list of alternatives based on the re-evaluated alternatives in the interim alternatives list and the additional alternatives it has produced. The stochastic input combiner 137 then chooses the top ranked alternatives for display in alternatives list 530.

Because of the operation of the natural language model, the alternatives list 530 is hopefully better than alternatives list 330 of FIG. 3. If that is the case, the user may choose an alternative 502 from alternatives list 530 without having to edit the text selection 310. In the example shown in FIG. 5, new text 504 is formed by the user replacing text selection 310 with alternative 502.

Flow Charts for a Typical Implementation

FIG. 6 is a flow chart of the steps in a typical routine 600 for processing source data. This routine illustrates the steps that implement the embodiments of the invention described with respect to FIGS. 3 5 and 11. The routine 600 begins at step 602 with the word processor receiving source data from multiple input sources. An input source may be a stochastic input source, such as handwriting recognition program module 230, speech recognition program module 240, or input method editor 250. An input source could also be non-stochastic, such as typed data entered using a keyboard 140 and mouse 142. Furthermore, some of the source data may come from two or more stochastic input sources configured in series. If data is derived from stochastic input sources configured in series, each stochastic input source may be counted as a different input source.

After the word processor receives source data from multiple input sources, the word processor combines that data into a multi-source text string in step 604. This means that the word processor creates text corresponding to the source data so that the word processor can display the text on the monitor 147. Furthermore, word processor creates a data structure to keep track of the source of each word of the text.

In step 606, a user may make a text selection including a portion of the displayed text. This text selection may include text from multiple input sources. The user may, for example, make the text selection by depressing a mouse button at the beginning of the text selection, dragging the mouse to the end of the desired text selection, and then releasing the mouse button. Preferably, the word processor highlights the text selection to indicate what has been selected.

In step 608, the word processor receives a "display alternatives" command for the text selection. In response, the word processor determines alternatives for the multi-source text selection in step 610.

In step 612, the word processor displays those alternatives on the monitor 147. The word processor preferably displays the alternatives in probabilistic order through a graphical user interface that allows the user to make a selection from the displayed alternatives. The graphical user interface may appear in a sub-window that the user can move around so as to reveal text hidden by the sub-window.

In step 614, the user gives the word processor a command. Examples of possible commands include the selection of a displayed alternative, an attempt to edit the text selection, or an attempt to make a new text selection by depressing the mouse button to anchor the mouse at a point of the text outside of the text selection.

In step 616, the word processor determines if the user has selected a displayed alternative. If the user has selected a displayed alternative, then step 618 is performed, in which the word processor replaces the text selection with the selected alternative. After step 618, the word processor discontinues the display of alternatives in step 624 before the routine ends at step 626. After step 626, the routine may be repeated by a return to step 602.

Returning to step 616, if the user has not selected a displayed alternative, then step 620 is performed. In step 620, the word processor determines if the user is editing the text within the text selection. If the user is editing the text within the text selection, then the word processor processes that edit in step 622. After the word processor completes the edit, the routine loops back

to step 610 to determine new alternatives for the edited text selection.

Returning to step 620, if the user command received in step 614 was not an edit command within the text selection, then the word processor performs step 624. In this case, the user has initiated the creation of a new text selection by depressing the mouse outside of the old text selection. Hence, the word processor discontinues the display of alternatives in step 624 before the routine ends at step 626. Once again, the routine may be repeated by returning to step 602.

FIG. 7 details the steps of routine 610 from FIG. 6. The routine describes the steps for determining alternatives for a multi-source text selection. Typically, a stochastic input combiner performs this routine. This combiner may be a program module in the word processor, a separate utility in the operating system, or a separate program that interfaces with the word processor.

The routine begins with step 702, in which the stochastic input combiner parses the text selection into text components originating from different input sources. To do so, the stochastic input combiner consults the data structure that stores the source of each word in the text string displayed on the monitor 147. Parsing the text selection into text components makes the process of determining alternatives for the text selection more manageable.

One skilled in the art should recognize that multiple definitions of a text component are possible. Using a different definition than the embodiment described in FIGS. 7-9 will require a different parsing step 702, as well as other appropriate modifications of the routines described therein. A text component could, for example, be a single word. Or, a text component could be

a phrase composed of words originating from the same input source. In the latter case, a phrase derived from a stochastic input source could be a different text component than an edit inserted into the middle of that phrase.

FIGS. 7 9 illustrate an example in which a text component is defined as the largest unit of text derived from a different input source or series of stochastic input sources than its neighbors, together with any typed text that has edited that text component. A unit of typed text that is not an edit of a text component originating from a stochastic input source is considered its own text component. For example, typed text a user has inserted between text components originating from different stochastic input sources is considered its own text component.

In step 704, the stochastic input combiner retrieves a stochastic model for each text component originating from a stochastic input source. In step 706, the stochastic input combiner determines if all stochastic models are lattices. If all stochastic models are lattices, then step 708 is performed.

In step 708, the stochastic input combiner concatenates all the lattices retrieved into a metalattice. In order to create the metalattice, the stochastic input combiner creates nodes to incorporate any typed text components for typed text that is not incorporated into a text component from a stochastic input source. Using lattice traversal methods well known to those skilled in the art, the stochastic input combiner traverses the metalattice to form a list of alternatives for the text selection in step 710. After step 710, control passes to step 716, which will be discussed after the "NO" path from step 706 is described.

Returning to step 706, if all stochastic models retrieved in step 704 are not lattices, then the stochastic input combiner performs step 712. In this case, at least one of the stochastic models is an "n-best" candidate list. So, the stochastic input combiner converts each of the lattices into an "n-best" candidate list. This is necessary for the stochastic input combiner to perform step 714.

In step 714, the stochastic input combiner combines the "n-best" candidate list for each of the text components with typed text components in order to form a combined list of alternatives for the text selection. The stochastic input combiner does this by forming all possible combinations of alternatives, one from each "n-best" candidate list for a text component. For each combination, the stochastic input combiner arranges the alternatives for the text components in the order in which the text components appear in the text selection. The list of all arrangements thus formed comprises the list of alternatives for the text selection. After step 714, control passes to step 716.

Once control passes to step 716 from either 710 or 714, the stochastic input combiner has formed a list of alternatives for the text selection. Step 716 is an optional step in which a natural language model is operative. If this step is performed, the stochastic input combiner submits the list of alternatives for the text selection to the natural language model.

If optional step 716 is performed, then optional step 718 is also performed. In that step, the natural language models returns a ranked list of revised alternatives to the stochastic input combiner. The revised list includes a re-evaluation of the probabilities of the alternatives in the list of alternatives submitted to the

natural language model. The revised list may also contain new alternatives for the text selection that the natural language model formed. The natural language model creates this revised list using natural language principles, including an analysis of grammatical and other language clues that the natural language model applies to the text selection as a whole.

In step 720, the stochastic input combiner chooses the top ranked alternatives for display. If optional steps 716 and 718 were performed, then the stochastic input combiner chooses these alternatives from the revised list received from the natural language model. If not, then the stochastic input combiner chooses these alternatives from the list of alternatives that were created by either step 710 or step 714. After step 720, the routine ends at step 722.

FIG. 8 shows the steps of routine 704 on FIG. 7. The routine illustrates the steps the stochastic input combiner follows to retrieve a stochastic model for each text component of the text selection originating from a stochastic input source. The routine begins at step 802, which effectively forms a loop for processing each text component. In step 802, the stochastic input combiner retrieves a text component. In step 804, the stochastic input combiner determines if that text component is a stochastic text component. If the retrieved component is not a stochastic text component, then step 806 is performed. In this case, the text component typically comprises typed text entered using a keyboard and mouse. Because the text component is non-stochastic, the stochastic input combiner assigns the text component a 100% probability. The stochastic input combiner then performs step 818, which shall be discussed shortly.

Returning to step 804, if the stochastic input combiner determines that the text component retrieved in step 802 is stochastic, then the stochastic input combiner performs step 808. In step 808, the stochastic input combiner determines if the text component is derived from stochastic models configured in series. If the text component is derived from a series of stochastic input sources, then the stochastic input combiner performs routine 810 in order to derive a series stochastic model that accurately represents the probabilities of the results produced by the last stochastic input source of the series. After routine 810, the stochastic input combiner performs step 812. Likewise, if the stochastic input combiner determines in step 808 that the text component retrieved in step 802 is not derived from a series of stochastic models, step 812 is performed.

In step 812, the stochastic input combiner determines if the user has edited the text component using a keyboard and mouse. If the text component has been edited, then the stochastic input combiner updates the corresponding stochastic model in step 814. If the stochastic model is a lattice, then updating it will include deleting any nodes corresponding to words that have been deleted from the text component. Furthermore, the stochastic input combiner must add nodes for new words within the text component. Similarly, if the stochastic model is an "n-best" candidate list, the stochastic input combiner must update each alternative of the list to remove words that have been deleted from the text component and add words that have been inserted into the text component.

After step 814, the stochastic input combiner performs step 816. The stochastic input combiner also performs step 816 if it determines in step 812 that the user has not edited the text

component. In step 816, the stochastic input combiner retrieves a stochastic model result for the text component that was selected in step 802. If the text component was derived from a series of stochastic models, then the stochastic model retrieved is the series stochastic model produced in step 810 or, if the text component has been edited, the series stochastic model that was updated in step 814. The stochastic model retrieved may be a lattice or an "n-best" candidate list. The stochastic model retrieved need only contain information about the selected text component, so the stochastic input combiner may retrieve the stochastic model from a larger stochastic model for a selection of text that includes the text component.

The text component that was selected in step 802 may be derived from stochastic input, but a stochastic model representing alternatives for that text component may be unavailable. In that case, the text component can be treated the same as a non-stochastic text component. In other words, the stochastic input combiner assigns the known alternative for the text component a probability of 100%. After step 816, the stochastic input combiner performs step 818.

Step 818 can be reached from either step 816 or step 806. In this step, the stochastic input combiner determines if there are any more text components in the text selection to process. If there are any more text components, then the routine loops to step 802 so the stochastic input combiner can get and process the next text component.

When there are no more text components to process in step 818, the stochastic input combiner optionally performs step 820 for incorporating the natural language model. In this step, the

stochastic input combiner passes each stochastic model retrieved for a text component to the natural language model. The natural language model applies natural language principles to the text components and returns them to the stochastic input combiner. Because the natural language model operates on individual text components in step 820, instead of on the entire text selection, step 820 may be performed either instead of steps 716 and 718, or in addition to those steps. After step 820, the routine ends at step 822.

FIG. 9 illustrates the steps of routine 810 on FIG. 8. This routine describes the steps the stochastic input combiner follows to derive a series stochastic model for a text component produced by stochastic input sources configured in series.

The routine 810 begins with step 902, which effectively begins a loop for processing in series order each of the stochastic input sources, except the last stochastic input source of the series. The first time the stochastic input combiner performs step 902, the stochastic input combiner selects the first stochastic input source in series order. This is the stochastic input source that receives the user input that ultimately produces the text component.

Because a stochastic input source produces multiple alternative results, the first stochastic input source produces multiple candidates for input into the second stochastic input source in the series. If the stochastic input combiner is not performing step 902 for the first time, then the stochastic input combiner will have produced a totalized candidate list in step 914 (to be described shortly) for the stochastic input source selected in step 902. In the latter case, the totalized candidate list contains the alternatives associated with the selected stochastic

input source that are to be used as input into the subsequent stochastic input source of the series. Step 904 effectively begins a loop for processing all of the candidates associated with the selected stochastic input source. In step 904, the stochastic input combiner retrieves one of the candidates for the selected input source.

In step 906, the stochastic input combiner submits the candidate retrieved in step 904 as input into the subsequent stochastic input source in series order. Inputting this candidate into the subsequent stochastic input source produces a stochastic result because the subsequent source is also stochastic. The stochastic input combiner retrieves this stochastic result.

In step 908, the stochastic input combiner determines if the stochastic result retrieved in step 906 is a lattice. If the stochastic result retrieved in step 906 is not a lattice, then it is a ranked candidate list and step 912 (to be discussed shortly) is performed. If the stochastic result retrieved in step 906 is a lattice, then the stochastic input combiner must convert the lattice into a ranked candidate list of alternatives, with each alternative having an associated probability. This is done in step 910 before control passes to step 912.

In step 912, the stochastic input combiner determines if there is another candidate for the selected source. If there is another candidate for the selected source, then the routine loops back to step 904 so that the stochastic input combiner can get the next candidate. If there is not another candidate for the selected source, then step 914 is performed.

In step 914, the stochastic input combiner combines all the candidate lists produced by using candidates from the input source

selected in step 902 as input into the subsequent stochastic input source in series order. This combination forms a totalized candidate list for the subsequent stochastic input source. The stochastic input combiner forms the totalized candidate list by making one entry for each unique candidate in the candidate lists being combined. Then, the stochastic input combiner calculates a probability for each alternative in the totalized alternatives list by summing all probabilities assigned to that alternative in each of the candidate lists being combined. The stochastic input combiner assigns each calculated probability to its associated alternative.

In step 916, the stochastic input combiner determines if there is another source in the series after what is currently the subsequent source. If there is another source in the series, then the stochastic input combiner selects what is currently the subsequent source in step 902, and the next source after what had been the subsequent source becomes the subsequent source. At this point, the stochastic input combiner chooses candidates from the totalized candidate list for the selected input source as input into the subsequent stochastic input source.

Returning to step 916, if there is not another source in the series after the subsequent source, then the routine ends at step 918. The totalized candidate list most recently created in step 914 is selected as the series stochastic model.

FIG. 10 is a logical flow diagram illustrating typical steps of an alternative embodiment 1000 of a source data processing routine. Generally, the routine provides for automatically adjusting the unit of text corrected in response to a user's text selection.

Routine 1000 begins with step 1005. In that step, the word processor 210 receives a text selection from the user which the user wants to correct. The user may specify the text selection by selecting the word or group of words comprising the text selection with the mouse 142. Alternatively, the user may specify a text selection consisting of a single word by using the mouse 142 to place the insertion point in or adjacent to the word. The word processor 210 may then submit the text selection to the stochastic input combiner 137 to determine correction alternatives.

In step 1010, the stochastic input combiner 137 submits the text selection to a correction scope model to determine if the scope of correction should be adjusted. Typically, adjusting the scope of correction involves identifying a text unit that will provide better text correction alternatives to the user than the text selection alone. For instance, the text selection may not include neighboring words that also contain errors which could be corrected together with the text selection. Usually, such errors in words neighboring a user's text selection are identifiable because they relate to errors in the text selection.

Accordingly, a text unit identified by the correction scope model may include the text selection plus one or more adjacent words. Instead of identifying only a single text unit for possible correction, the correction scope model may identify multiple text units, each of which are likely to yield good alternatives for text correction.

In step 1015, the stochastic input combiner 137 receives from the correction scope model a list of text units for which correction alternatives should be provided to the user. If the correction scope model determined that the scope of correction need not be

adjusted, then the list of text units includes only the text selection. If the correction scope model identified only one text unit for correction, the list of text units need include only that one text unit. Step 1020 effectively begins a loop for processing each of the text units identified in the list of text units that the correction scope model returned to the stochastic input combiner 137 in step 1015. In step 1020, the combiner 137 selects a text unit for processing. In step 1025, the combiner 137 performs the steps of the routine of FIG. 7 in order to determine alternatives for the selected text unit. One should understand that when the combiner 137 performs the routine 1025 by performing the steps described in FIG. 7, "text selection" as used in FIG. 7 refers to the selected text unit.

In step 1030, the stochastic input combiner 137 determines if there are any more text units to process. If there are more text units, the routine loops back to step 1020 along the "YES" branch to process the next text unit. If there are no more text units, the "NO" branch is followed to step 1035.

In step 1035, the stochastic input combiner 137 provides each of the correction alternatives and their associated text units to the word processor 210 for display. The word processor 210 may display these alternatives in any appropriate manner. If the scope of correction was not adjusted, the alternatives may be displayed as described with respect to FIG. 6. If the combiner 137 expanded the scope of correction to a single text unit, the word processor 137 may highlight the additional words in the text to which the scope of correction was expanded in a different color than the color used to highlight the text selection, and the word processor may present the

alternatives for the text unit in a typical graphical user interface as described with respect to FIG. 6.

Suppose that the correction scope model identified multiple text units for correction. In that case, the word processor 210 may present the user with a menu of alternatives and identify the corresponding text unit for each alternative.

After the word processor 210 presents the correction alternatives to the user through a graphical user interface, the routine ends at step 1040. The word processor 210 can then process the user's response to the alternatives as described with respect to FIG. 6.

Conclusion

Other alternative embodiments will become apparent to those skilled in the art to which an exemplary embodiment pertains without departing from its spirit and scope. Accordingly, the scope of the present invention is defined by the appended claims rather than the foregoing description.

4 Brief Description of Drawings

FIG. 1 is a block diagram illustrating the operating environment for an exemplary embodiment of the present invention.

FIG. 2 is a block diagram providing an overview of the program modules of a multi-source data processing system.

FIG. 3 is a block diagram that illustrates the operation of a typical embodiment of the present invention.

FIG. 4 is block diagram that illustrates the operation of an embodiment of the present invention that allows the user to edit a text selection.

FIG. 5 is a block diagram that illustrates another embodiment of the present invention in which a natural language model is operative.

FIG. 6 is a flow chart of the steps in a routine for processing source data from multiple input sources.

FIG. 7 is a flow chart of the steps in a routine for determining alternatives for a multi-source text selection.

FIG. 8 is a flow chart of the steps in a routine for retrieving stochastic models for the text components in a text selection.

FIG. 9 is a flow chart of the steps in a routine for deriving a series stochastic model.

FIG. 10 is a flow chart of the steps in an alternative routine for processing source data that includes changing the scope of correction of a text selection.

FIGS. 11 is block diagram that illustrates the operation of an embodiment of the present invention that allows the user to edit a text selection.

[図 1]

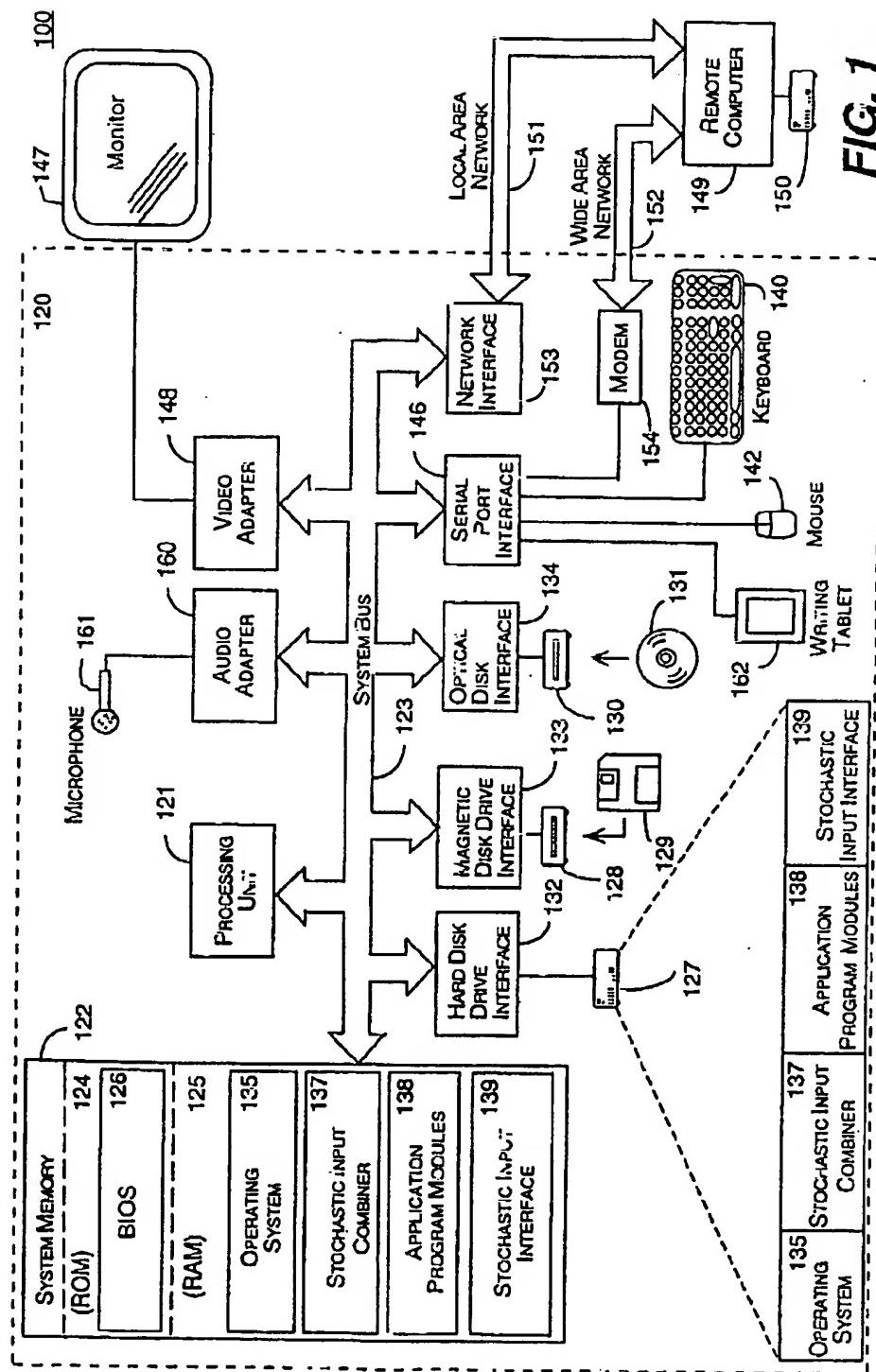


FIG. 1

[図2]

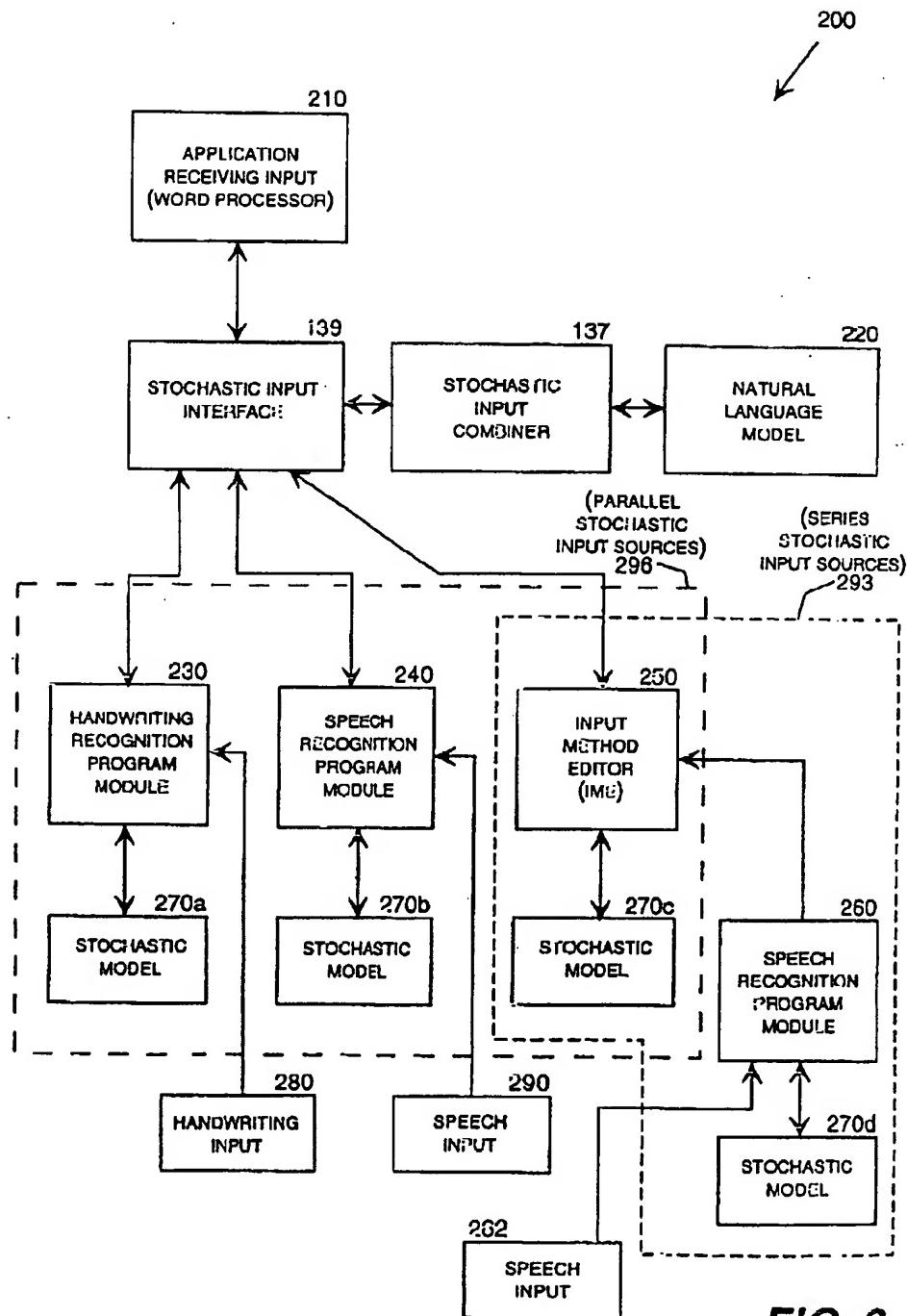


FIG. 2

【図3】

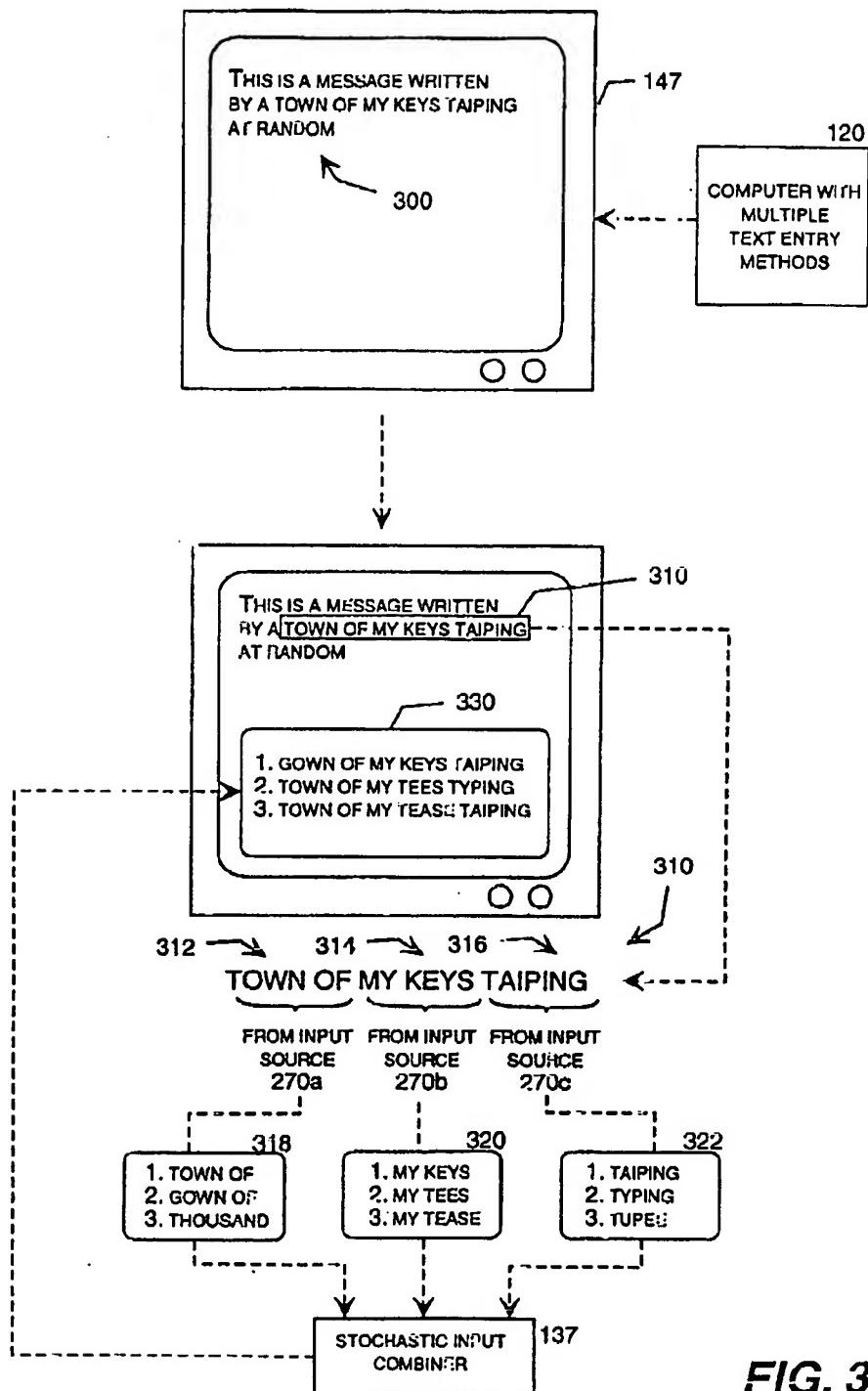
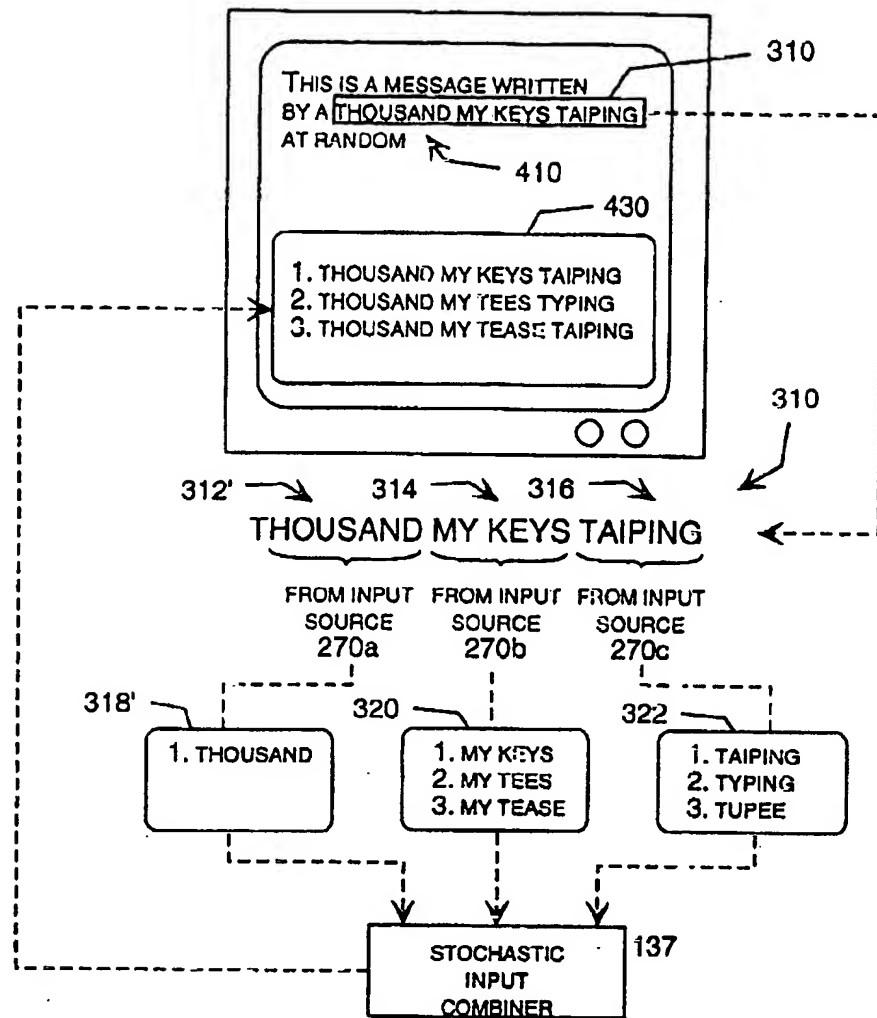


FIG. 3

[図4]



[FIG. 5]

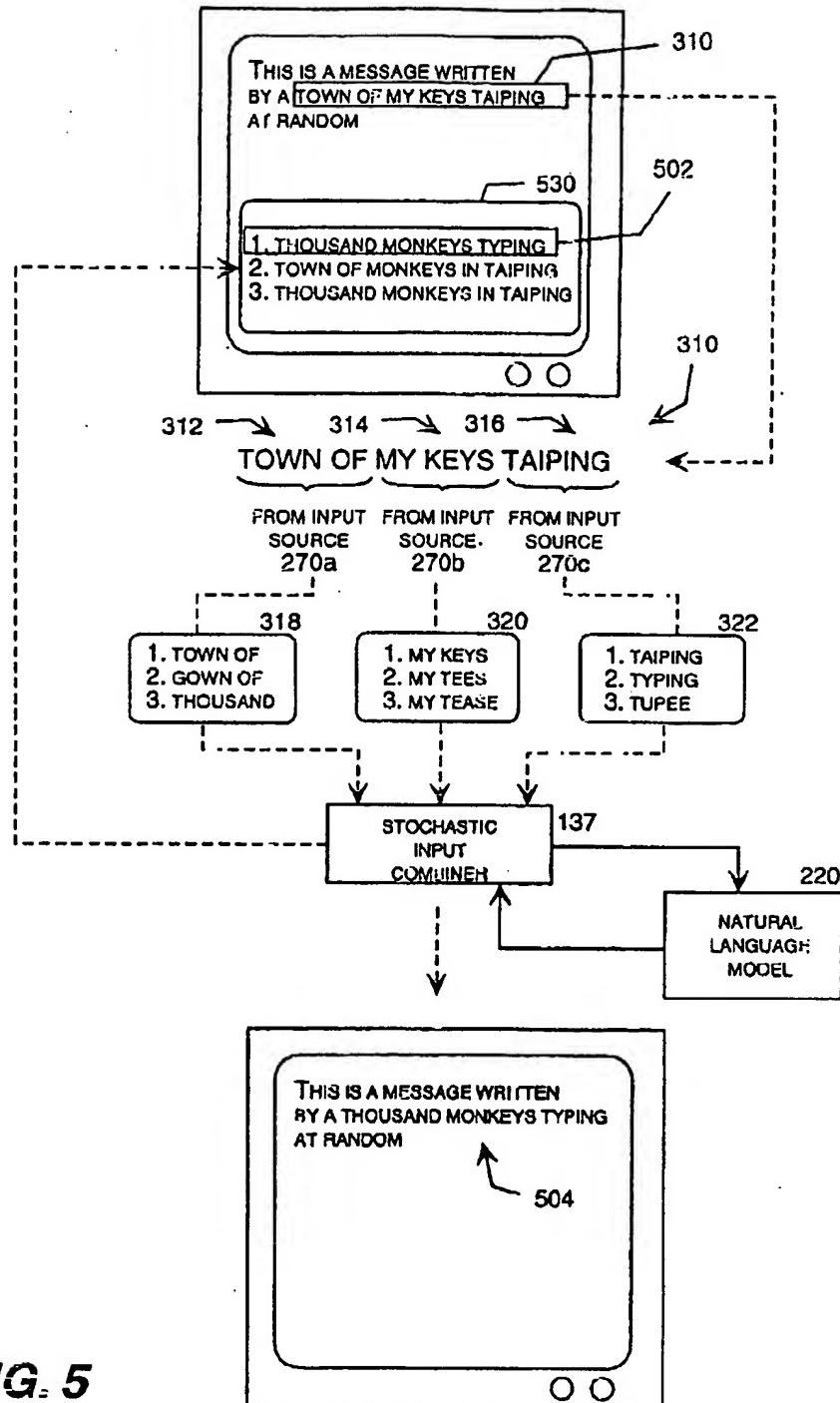
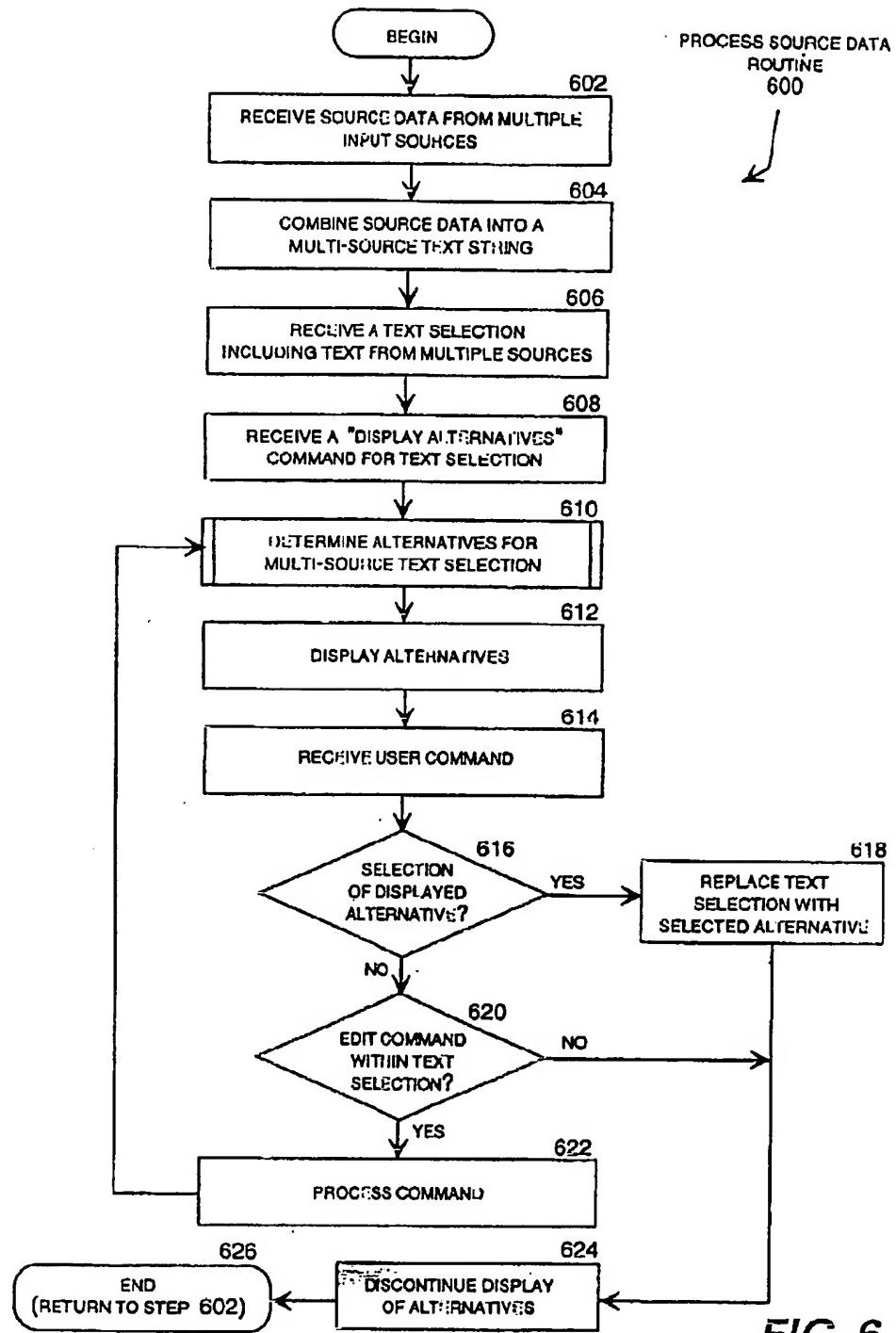


FIG. 5

【図6】



[図7]

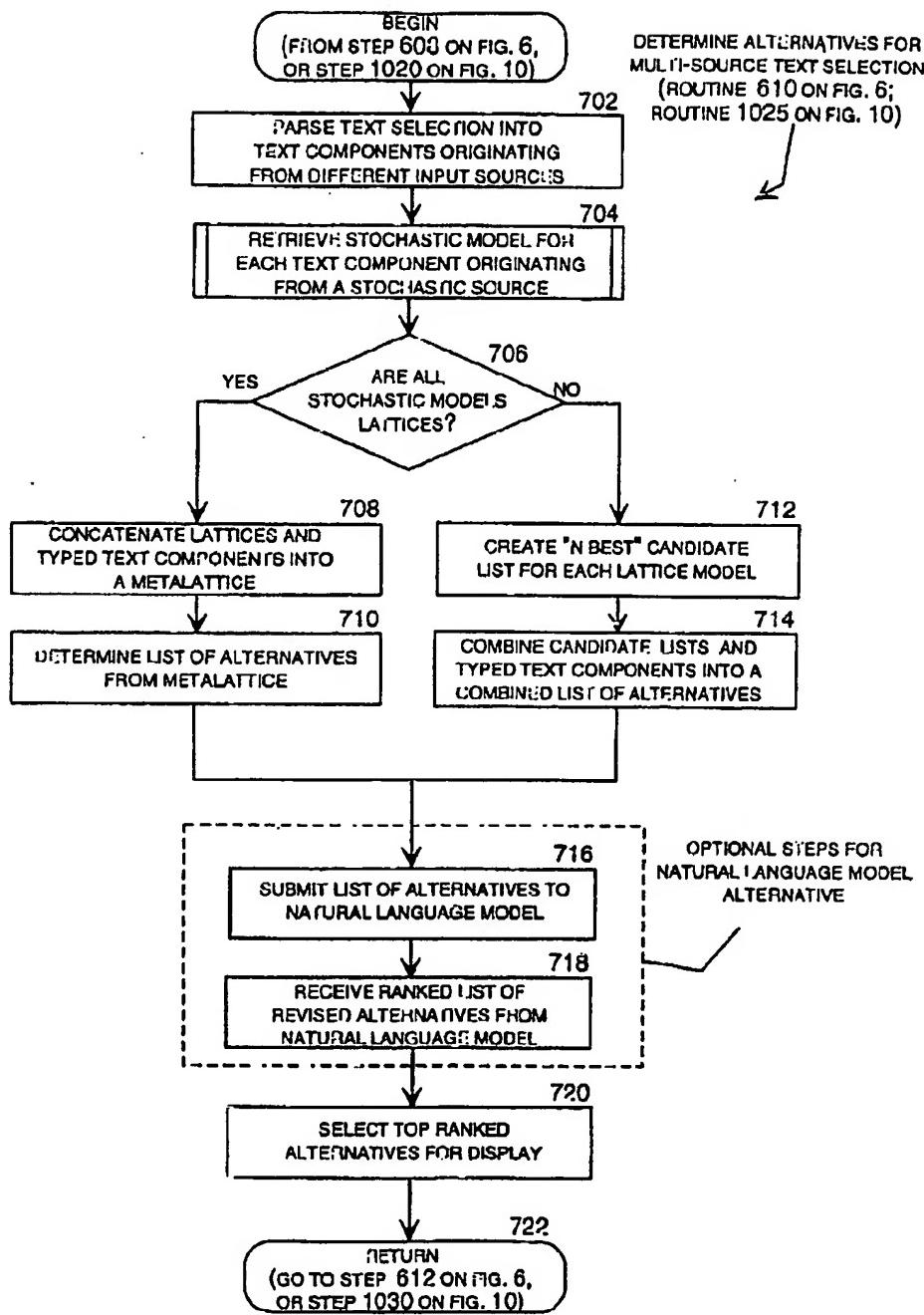


FIG. 7

[図8]

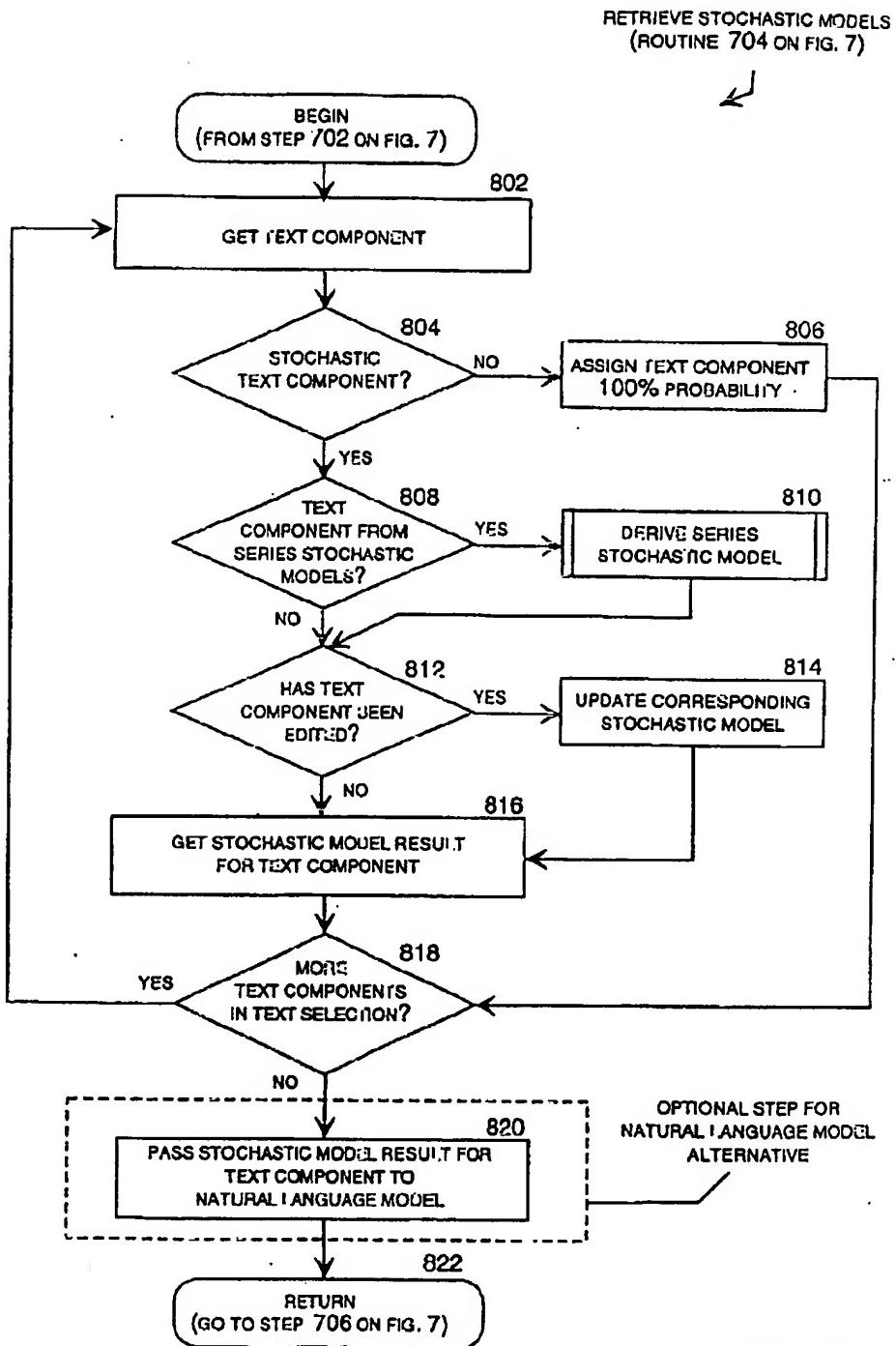


FIG. 8

[FIG 9]

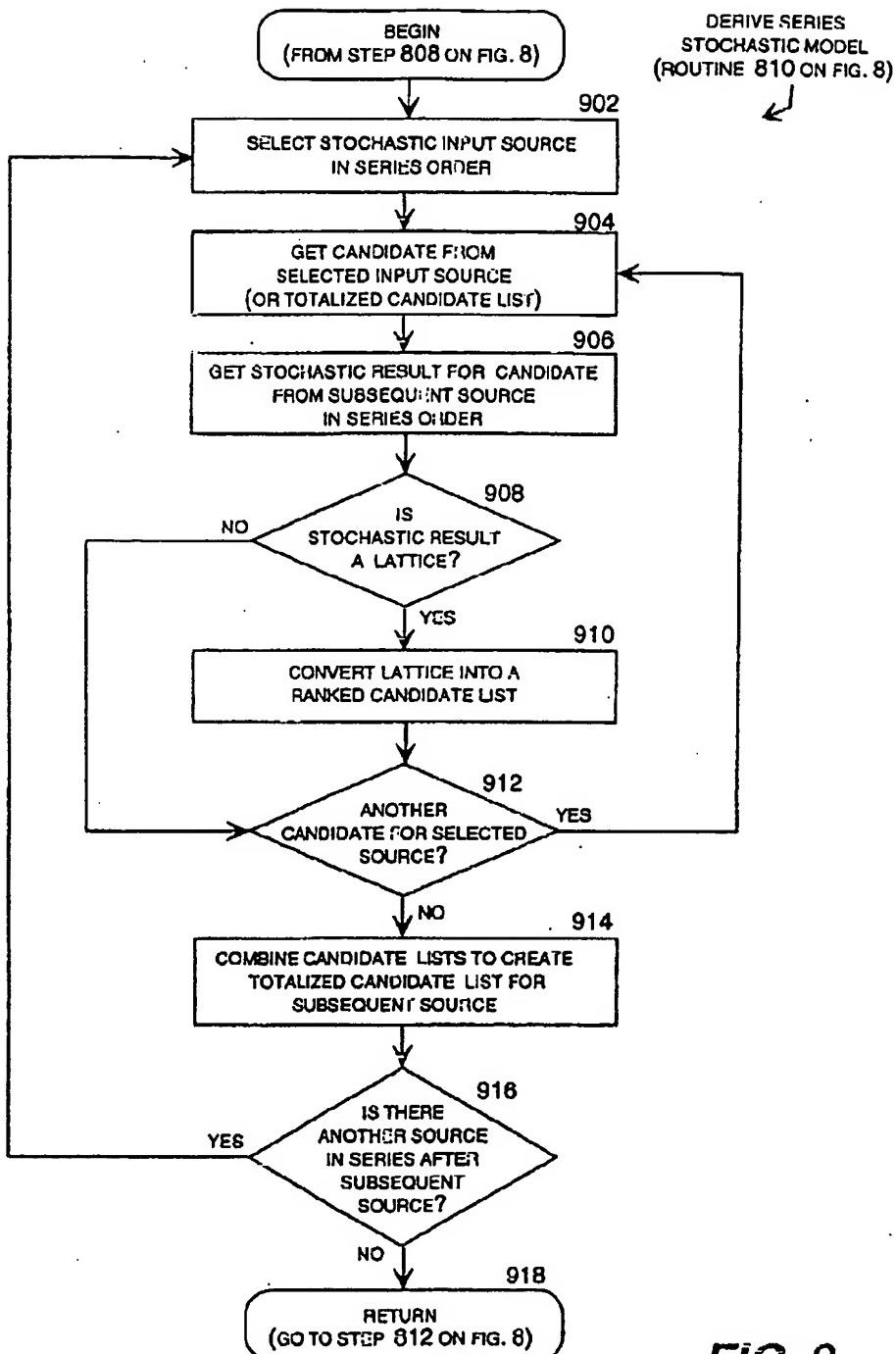


FIG. 9

[FIG 10]

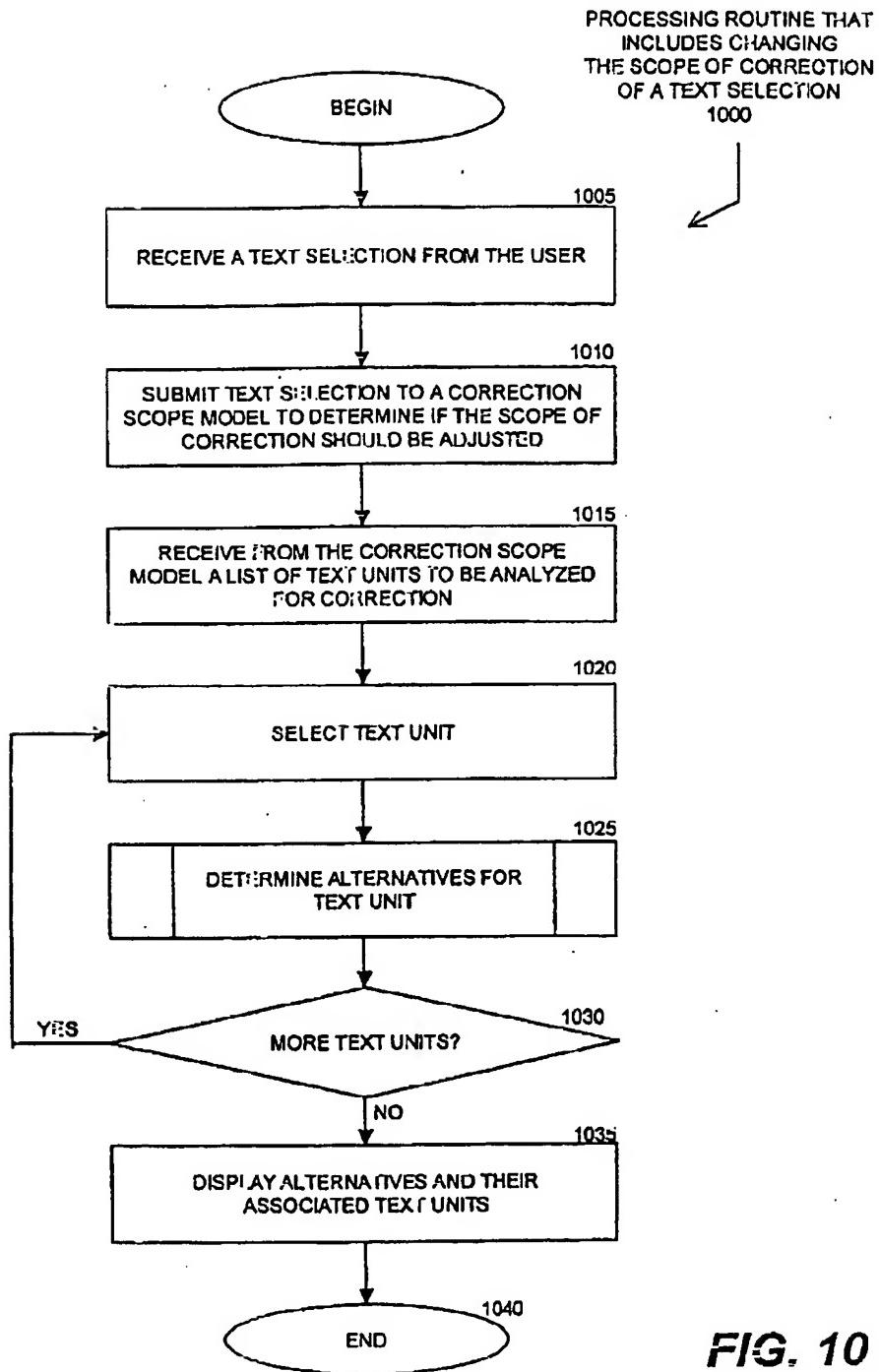
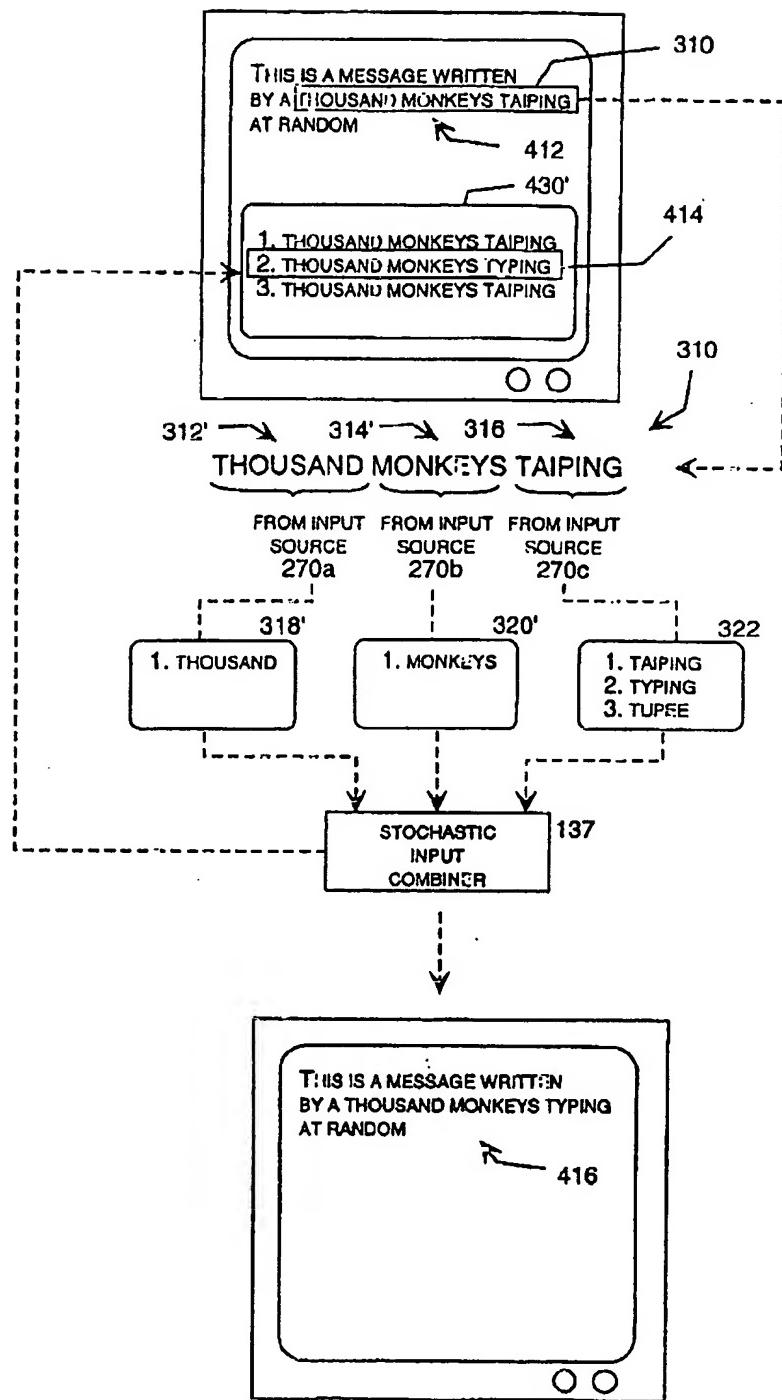


FIG. 10

[図11]



1 Abstract

A computer-implemented method for providing a candidate list of alternatives for a text selection containing text from multiple input sources, each of which can be stochastic (such as a speech recognition unit, handwriting recognition unit, or input method editor) or non-stochastic (such as a keyboard and mouse). A text component of the text selection may be the result of data processed through a series of stochastic input sources, such as speech input that is converted to text by a speech recognition unit before being used as input into an input method editor. To determine alternatives for the text selection, a stochastic input combiner parses the text selection into text components from different input sources. For each stochastic text component, the combiner retrieves a stochastic model containing alternatives for the text component. If the stochastic text component is the result of a series of stochastic input sources, the combiner derives a stochastic model that accurately reflects the probabilities of the results of the entire series. The combiner creates a list of alternatives for the text selection by combining the stochastic models retrieved. The combiner may revise the list of alternatives by applying natural language principles to the text selection as a whole. The list of alternatives for the text selection is then presented to the user. If the user chooses one of the alternatives, then the word processor replaces the text selection with the chosen candidate.

2 Representative Drawing

Fig. 1